Interactive project review of deformable parts through haptic interfaces in Virtual Reality

Zhaoguang Wang

N° d'ordre : 4367 ANNÉE 2011

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Mécanique

Ecole doctorale MATISSE

présentée par

Zhaoguang WANG

préparée à l'unité de recherche (IRISA UMR6074)
Institut de Recherches en Informatique et Systèmes Aléatoires
Université de Rennes 1

Interactive project review of deformable parts through haptic interfaces in Virtual Reality

Thèse soutenue Rennes le 22/06/2011

devant le jury composé de:

Isabelle PUAUT
Professeure Université de Rennes 1

Jean-Claude LÉON
Professeur ENSE3-Grenoble-INP,
Laboratoire G-SCOP

Xavier FISCHER
Professeur ESTIA/HDR

Judy VANCE
Professor of Iowa State University, USA,
Virtual Reality Applications Center (VRAC)

David DEFIANAS
PSA Peugeot Citroën,
Centre Technique de Vélizy (VVB)

Georges DUMONT
Maître de Conférences/HDR,
ENS Cachan-IRISA-INRIA,
Directeur de thèse

Acknowledgments

I would like, first, to express my gratefulness to my supervisor Georges Dumont for his support and guidance throughout my work, friendliness, patience and insightful comments.

Sincere thanks to all the jury members. Thank you for your time to review the thesis.

My thanks to all the team members with whom I spend a memorable time. Thank you very much.

I also want to thank my parents and my elder sister who always stay beside me, with priceless supports.

At last, we want to say “thank you” to Ying for her love.

Abstract

PHysical prototypes are increasingly replaced by virtual prototypes in the industrial implementation of Product Lifecycle Management. The design evaluation of an industrial deformable mechanical part plays a crucial role in term of validating its functional properties. From the industrial point of view, a deformable model formulated by the Finite Element Method is normally employed. However, the employment of the model is not straightforward for real-time interactions, especially when haptic interfaces are introduced into these deformation evaluation applications. Recently, a pre-computation approach based on the model reduction method was commonly used to reduce the real-time computational loads.

The main goal of this thesis is to extend the pre-computation approach toward the design validation of deformable mechanical parts to investigate the trade-off issue between the deformation accuracy and the interaction performance. The key idea is to conceive techniques treating the off-line pre-computations and the on-line haptic interactions. Particularly, we develop a real-time deformation simulation framework by proposing a two-stage method combining an off-line phase and an on-line phase. During the off-line phase, we compute deformation spaces based on the modal analysis. The off-line pre-computations contribute to the modelling of a costless real-time deformation model which is suitable for haptic interactions. Furthermore, we propose an off-line mesh analysis method to pre-compute modal deformation spaces regarding the anticipated deformation evaluation scenarios. A real-time switch among these different spaces is developed so that the on-line deformation computations can focus on degrees of freedom where are necessary. During the on-line phase, we divide the real-time deformation computation process into two separate modules which are implemented on different threads to ensure the real-time haptic interaction performance. One module is dedicated to the haptic update task, which is implemented by extracting a sub-matrix from the pre-computed modal matrix, while the other module is dedicated to the deformation computation and visualization task.

To verify the proposed method in the thesis, we carry out interaction experiments by interacting with different models with an increasing complexity. Experimental results show that our method can efficiently handle the trade-off issue, as the deformation modelling is formulated by the finite element method which guarantees the deformation accuracy. And moreover, the heavy computations of large elastic systems are occurred off-line which assure a costless deformation response model in real-time.

Keywords: Virtual Prototyping, Deformation Verification, Finite Element Method, Model Reduction, Modal Analysis, Haptic Interaction, Virtual Reality.

Résumé Etendu

L'avènement de la technologie informatique a profondément modifié tous les aspects de la vie quotidienne ou professionnelle. Plus particulièrement, le traitement de l'information a bénéficié de ces immenses progrès. Dans cette thèse, nous proposons de nous concentrer sur le cas plus précis de la Gestion du Cycle de vie d'un Produit ("Product Lifecycle Management" ou PLM), qui est traité par des technologies de l'information rigoureuses. Ainsi, le PLM fournit des outils pour définir ou concevoir des produits. Il regroupe toutes les informations liées aux produits ainsi que les échanges d'informations entre les différents acteurs de leur cycle de vie. Parmi ces outils, l'Ingénierie Assistée par Ordinateur (IAO) propose de créer et de modifier le produit dans un format numérique. Par ailleurs, elle fournit des méthodes de simulation numérique pour l'analyse des fonctionnalités du produit. Un des avantages principaux d'un tel processus numérique est la réduction du temps de conception et la réduction des coûts induits lors des modifications des produits. L'échange des données de produits au sein du PLM est également un atout.

Des outils de simulation sont disponibles en IAO. Ils permettent l'évaluation de fonctionnalités du produit. L'inconvénient principal de ces simulations réside dans deux principaux aspects. D'une part, la mise en œuvre d'une simulation d'IAO, par exemple une simulation par la méthode des éléments finis (FEM), est un processus d'autant plus long que le concepteur souhaite un résultat précis, surtout dans les cas où l'on souhaite simuler une partie mécanique complexe d'un système industriel. D'autre part, les modes d'interaction au cours de ces simulations sont relativement pauvres. Par exemple, les ingénieurs ne peuvent normalement pas accéder à des résultats de simulations intermédiaires pour régler les paramètres de simulation de manière interactive. Ces difficultés diminuent l'efficacité des flux d'information et diminuent l'apport du processus du PLM.

Dans un autre domaine, le développement des technologies de l'information stimule l'invention de solutions nouvelles en utilisant des équipements techniques avancés qui amènent les opérateurs humains au plus près des données scientifiques de différents domaines. Par exemple, la Réalité Virtuelle (RV) est un domaine prometteur dans lequel un opérateur est immergé dans l'espace produit. Cet espace produit est caractérisé par des rendus réalistes et multi-sensoriels et des capacités d'interaction enrichies. Ainsi, la technologie utilisée en Réalité Virtuelle révèle une grande variété d'applications potentielles qui vont de l'exploration massive des données scientifiques, à la formation chirurgicale et au prototypage virtuel.

Le travail présenté dans cette thèse s'inscrit dans le cadre de l'évaluation industrielle des maquettes numériques représentant des objets déformables dans un environnement de réalité virtuelle en introduisant l'homme, en tant qu'opérateur, dans la boucle. Il vise donc à l'intégration du prototypage virtuel dans un PLM. La validation interactive de la conception de pièces mécaniques, en se basant sur les pratiques industrielles, joue un rôle important dans le processus de conception, car une simulation interactive de

déformation dans un environnement de réalité virtuelle permet aux ingénieurs des différents secteurs industriels “d’entrer” dans l’écran pour manipuler ces maquettes numériques. Cette manipulation permet d’envisager l’identification de problèmes très tôt dans la phase de conception. De cette façon, le temps et les coûts requis pour le partage des informations entre les différents secteurs seraient considérablement réduits, et donc l’efficacité de cet échange d’informations de conception au sein du PLM pourrait être considérablement augmentée.

Toutefois, une simulation interactive de déformation dans un environnement de réalité virtuelle à travers des interfaces haptiques est une tâche difficile induite. En effet, nous devons faire face à la problématique de gestion du compromis entre la précision de calcul du problème de déformation et la performance de la simulation qui permet l’interaction en temps-réel. Ce problème est particulièrement présent dans les cas complexes de simulation où des pièces déformables sont prises en compte. En effet, le degré de réalisme pour la représentation des déformations tend à exiger une haute résolution de maillage qui, à son tour, a un impact direct sur la taille des matrices impliquées dans la simulation et engendre des calculs volumineux difficiles à réaliser en temps réel.

Récemment, une approche de pré-calcul basée sur la méthode de réduction de modèle a été proposée pour l’interaction haptique en temps-réel avec des objets déformables. L’objectif principal de cette thèse est d’étendre une telle approche de pré-calcul dans le cas de la validation de la conception de pièces mécaniques déformables en traitant le problème du compromis entre les questions d’exactitude de déformation et de la performance d’interaction. L’idée principale est de proposer des méthodes de traitement hors-ligne basées sur des pré-calculs des phénomènes physiques et des méthodes d’interaction haptique en ligne. En particulier, nous proposons une méthode de simulation des déformations en temps-réel en introduisant une méthode en deux étapes, associant une phase hors ligne et une phase en ligne. D’une part, au cours de la phase hors ligne, nous (pré-)calculons une base de l’espace des déformations en utilisant une analyse modale. Ces pré-calculs hors-ligne permettent la modélisation de la déformation pour une utilisation (en ligne) en temps-réel et compatible avec la modalité d’interaction haptique. En outre, nous proposons une méthode d’analyse de maillage hors-ligne pour affiner le pré-calcul des espaces modaux de déformation. Cette analyse des maillages est guidée par les scénarios industriels d’évaluation de conception. Ces scénarios sont basés sur une expertise “métier”. La méthode que nous proposons permet de commuter en temps-réel entre ces différents espaces de sorte que les calculs de déformation en ligne peuvent se concentrer sur certains degrés de liberté et ainsi améliorer la précision des calculs sans en diminuer la performance. D’autre part, au cours de la phase en ligne, nous divisons le processus de calcul en temps-réel de la déformation en deux modules distincts qui sont mis en œuvre sur différents processus et permettent d’assurer l’exécution de l’interaction haptique en temps-réel. Le premier module est consacré à la tâche de mise à jour de la partie haptique. Elle est mise en œuvre par l’extraction d’une sous-matrice à partir des données pré-calculées initialement. Le second module est consacré au calcul de la déformation au cours de l’interaction et au calcul de la tâche de visualisation.

Nous présentons plus de détails sur la structure globale de ce travail de thèse dans la suite de ce résumé étendu.

Chapitre 1

Dans ce chapitre, nous présentons d'abord le rôle important du PLM dans un processus industriel de conception, dans l'évolution de ce processus ainsi que les avantages qu'apporte la représentation sous forme numérique d'un produit. Le travail présenté dans cette thèse s'inscrit dans le cadre qui consiste à employer les technologies de la réalité virtuelle pour l'évaluation de déformation en temps-réel et qui pourrait constituer un des éléments clés du PLM appliqué à la conception de pièces mécaniques déformables. Le PLM en conception de produit est très implanté dans l'industrie automobile, car il permet de gérer le produit conservant et en échangeant les informations qui relatives à tous les stades de son existence. Ces informations sont échangées entre les différents acteurs dans l'ensemble du processus dans le PLM. Un meilleur suivi d'un produit dans l'entreprise fournit une plus grande continuité d'information, ainsi qu'une meilleure conformité entre la conception effectuée en bureau d'études et la réalisation effective sur la chaîne de production. De manière générale, la gestion d'un produit peut couvrir toutes les étapes d'une production ou une partie de celle-ci: la phase de conception, la phase de conception du modèle CAO et la phase de l'évaluation, la phase de fabrication et la phase d'exploitation du produit. Nous nous concentrons sur le processus d'évaluation de pièces mécaniques industrielles en cours de conception (voir la Figure 1.1), tandis que les autres composantes du PLM ne sont pas l'objet de cette thèse.

En second point, dans le contexte de la mise en œuvre du PLM, les simulations numériques doivent être effectuées avant d'entrer dans le processus de fabrication. Les tâches principales pour les ingénieurs de conception sont de mettre en œuvre les processus de validation qui permettent de satisfaire les exigences de conception et les contraintes des conditions de travail réelles. De tels résultats de simulation sont utiles à tous les stades de la conception des produits. Toutefois, les principaux inconvénients des simulations basées sur des outils traditionnels résident dans deux aspects. D'une part, la mise en œuvre d'une simulation, par exemple d'analyse par éléments finis, tend à être un processus fastidieux pour parvenir à un résultat d'analyse souhaitable, en particulier quand la complexité de la pièce mécanique augmente. D'autre part, les modes d'interaction au cours de ces simulations sont relativement pauvres. Par exemple, les ingénieurs n'ont pas accès aux résultats intermédiaires de la simulation pour régler les paramètres de simulation de manière interactive. Ces inconvénients des simulations classiques diminuent l'efficacité du processus du PLM. En revanche, la réalité virtuelle est un domaine dans lequel on cherche à plonger un opérateur dans l'espace produit en lui offrant un environnement naturel et des méthodes d'interaction intuitives.

Troisièmement, nous proposons une étude d'applications qui tendent à démontrer le potentiel de la réalité virtuelle. Nous nous concentrons sur l'intégration des applications existantes de RV-PLM dans le contexte industriel en classant ces applications à partir de quatre thèmes différents:

- RV pour des application de montage/démontage,
- RV pour la revue de projets,
- RV pour l'analyse de l'ergonomie
- RV pour la simulation interactive de déformation.

Toutes ces applications partagent un objectif commun de réduire les coûts de mise en œuvre d'un système PLM. Par ailleurs, l'emploi généralisé d'interfaces haptiques dans des environnements de Réalité Virtuelle améliore non seulement la performance des utilisateurs dans la réalisation de tâches virtuelles, mais offre également la possibilité d'enrichir les canaux d'interaction durant les différentes phases de conception de la pièce. Une grande majorité de ces applications industrielles d'évaluation se concentrent sur des opérations d'assemblages ou de maintenance basées sur une interaction haptique. Les demandes concernant l'analyse interactive par éléments finis dans un environnement de réalité virtuelle sont peu explorées.

Quatrièmement, nous présentons les motivations de la thèse à partir de deux aspects. D'une part, l'interaction en temps réel haptique avec des modèles déformables soulève la question du compromis entre l'exactitude du calcul de la déformation et la performance de l'interaction. Un premier aspect est que le taux de rafraîchissement nécessaire pour la boucle de rendu haptique soit au moins égal à 1000 Hertz pour assurer l'interaction de l'utilisateur. L'autre aspect est que les modèles déformables basés sur la physique (mécanique) du phénomène sont les plus à même de représenter des résultats de déformation fidèles à la réalité. Les relations de la théorie de l'élasticité sont généralement utilisées pour établir la formulation mathématique du problème, qui est finalement résolue par la méthode des éléments finis (FEM). Effectuer des calculs FEM au sein d'une boucle de rendu haptique dépasse généralement les capacités de calcul des ordinateurs. La méthode de simulation des pièces déformables proposée dans cette thèse est consacrée à un cas de valorisation industrielle qui s'inscrit dans le projet EMOA[1] piloté par le groupe PSA Peugeot Citroën et financé dans le cadre des pôles de compétitivité. Plus particulièrement, le fondement de cette thèse concerne l'évaluation de la déformation de pièces mécaniques déformables. Cela implique, par exemple, la vérification de la conformité aux spécifications pour des pièces représentées par des modèles CAO de plus en plus complexes. En ce qui concerne le processus d'emboutissage dans l'industrie automobile, la vérification de la conformité d'un polystyrène, à partir de laquelle le moule de l'outil d'emboutissage sera réalisé, est de première importance. Comme les processus de vérification de déformation peuvent être différents selon qu'ils sont réalisés en la réalité ou en environnement virtuel, nous définissons des scénarios de vérification de déformations locaux et globaux (voir Figure 1.9 et la Figure 1.10) dans le but d'établir les paradigmes d'interaction correspondants dans un environnement de RV.

Enfin, le champ d'application et les contributions de la thèse sont brièvement décrits. Les détails de la déformation en deux étapes sont présentés par la suite.

Chapitre 2

Sur la base de la discussion du premier chapitre, une méthode de modélisation des déformations doit être proposée pour représenter les modèles industriels déformables. Dans ce chapitre, nous décrivons les connaissances théoriques nécessaires à la modélisation de la déformation, y compris la théorie de l'élasticité, la formulation de la méthode des éléments finis ainsi que la méthode de réduction de modèles. Enfin, un bref état-de-l'art sur les logiciels d'éléments finis est présenté, car les méthodes de modélisation des déformations basées sur des pré-calculs reposent fortement sur la méthode

[1] EMOA: Excellence Dans la Maîtrise de l'Ouvrant Automobile.

des éléments finis dont les codes sont disponibles à titre gratuit ou à titre commercial.

Tout d'abord, les méthodes de modélisation des déformations peuvent être classées en deux catégories principales qui sont désignées comme des méthodes non-basées sur la physique et les méthodes basées sur la physique. Les méthodes non-physiques pour la modélisation des objets déformables sont généralement basées sur des techniques géométriques heuristiques ou appliquent des principes physiques simplifiés pour obtenir des résultats de visualisation plausibles. Ils ont été très populaires principalement dans l'infographie. Les méthodes non-basées sur la physique sont généralement rapides et efficaces pour représenter l'animation interactive des déformations. Toutefois, ils ne sont pas appropriés pour la modélisation des cas où le comportement de la déformation physique réaliste est souhaitée. Les méthodes de modélisation non-basées sur la physique ne sont pas adéquates pour décrire les propriétés des matériaux internes. Dans cette thèse, nous nous concentrons sur la validation de conception de pièces mécaniques déformables à travers des interfaces haptiques et nous nous baserons sur des méthodes qui s'appuient sur la représentation physique des phénomènes. Ces méthodes à base de modèles physiques sont fondées sur les modèles mathématiques, généralement formulés par un ensemble d'équations aux dérivées partielles (EDP), qui sont dérivées de la théorie de l'élasticité. Par conséquent, les méthodes de modélisation de la déformation à base physique sont capables de représenter les comportements des pièces mécaniques déformables. Malgré ce point, les calculs associés sont coûteux et peuvent devenir un obstacle à leur emploi pour le calcul de déformations en temps-réel.

Deuxièmement, la théorie de l'élasticité est généralement appliquée pour représenter les modèles de déformation à base physique formulées par la FEM. En nous basant sur les scénarios locaux et globaux de vérification de déformation suggérés par l'application industrielle, nous nous concentrons dans cette thèse, sur la représentation de petites déformations globales associées à de grands mouvements de corps rigides. L'approximation des déformations linéaires est utilisée ici pour décrire les petites déformations. Bien que cette approximation ne représente pas correctement les rotations finies, il peut contribuer à la réalisation d'applications haptiques en temps-réel tout en préservant une bonne représentation des petites déformations. (1) L'hypothèse de déformation linéaire permet des pré-calculs qui fournissent une représentation appropriée pour les interactions haptiques. Cela est particulièrement important pour l'utilisation de la réalité virtuelle dans le domaine du prototypage virtuel dans la phase de conception. En effet, au cours de cette phase de conception, les comportements de déformation non-linéaires ne sont pas nécessairement requis. (2) En ce qui concerne de nombreux types de modèles non-linéaires élastiques, les pré-calculs nécessaires pour les simulations de déformations en temps réel semblent difficile à proposer et il est difficile de prédire les situations d'interaction a priori. Pour les modèles élastiques linéaires, l'indépendance des pré-calculs au paramètre temps et la possibilité d'appliquer le des principe de superposition sont deux point qui rendent la méthode adaptée aux calculas de déformation en temps réel. La méthode en deux étapes proposée dans cette thèse tire avantage de cette hypothèse de linéarité des modèles déformables pour satisfaire les exigences d'évaluation des modèles industriels.

Troisièmement, nous présentons brièvement le processus de discrétisation des solides continus, y compris la génération de maillage, et l'écriture des équations du mouvement. Pour les applications haptiques, un emploi explicite de la FEM conduit généralement à des calculs volumineux qui empêchent l'interaction en temps réel. Une méthode

simplifiée est basée sur l'hypothèse des petites déformations en employant la théorie de l'élasticité linéaire, qui conduit à système linéaire d'équation , et ce qui permet un processus efficace de pré-calcul, une clé pour garantir les performances d'interaction. Pour la validation de déformations de pièces mécaniques complexes, la résolution d'équations différentielles dépendant du temps n'est pas nécessaire. Pour préserver le taux d'interaction haptique tout en conservant les avantages, en termes de précision notamment, des modèles aux éléments finis, nous proposons une méthode en deux étapes, associant une phase hors-ligne de pré-calcul et une phase d'interaction en ligne. Notre méthode de modélisation des déformations tire profit de la linéarité des équations pour effectuer un processus de pré-calcul basé sur la méthode de réduction de modèles. L'idée clé de l'application de la méthode de réduction dimensionnelle de modèle pour la simulation des déformations est de simplifier le système dynamique représenté par une série d'équations différentielles. Par cette méthode, la dimension du problème éléments d'origine peut être considérablement réduite, ce qui donne un système composé de moins d'équations différentielles et de moins de variables inconnues. Ces équations peuvent être résolues plus rapidement, en temps-réel, tout en contrôlant la perte de précision de la solution. La question clé de la méthode de réduction de modèle pour la simulation de la déformation est de choisir un sous-espace dans lequel la déformation peut être approchée correctement.

Enfin, puisque notre phase de pré-calcul hors-ligne est basée sur un générateur de maillage open-source et sur code éléments finis, nous présentons un bref aperçu sur les mises en œuvre actuelles des méthodes employées par la modélisation par éléments finis.

Chapitre 3

Dans ce chapitre, nous présentons les fondements de l'haptique dans un environement de réalité virtuelle puis nous présentons des travaux dédiés à la gestion du compromis classique entre l'exactitude de représentation des déformations et la performance du calcul de ces mêmes représentation.

Comme nous l'avons décrit dans le chapitre précédent, l'emploi des interfaces haptiques a été reconnu comme un élément essentiel pour l'intégration VR-PLM, étant donné que ces interfaces fournissent un canal intuitif pour les interactions entre les ingénieurs et les simulations dans un environnement virtuel. Nous présentons, en premier lieu, la classification de l'haptique dans un environnement de réalité virtuelle: l'haptique pour l'homme conduit à des études de perception, les interfaces haptiques qui proposent de rendre les sensation kinesthésiques et l'haptique du monde de la simulation qui traite des paradigmes d'interaction et du rendu haptique. Cette thèse se concentre sur le troisième domaine du rendu haptique et contribue à essayer de rendre haptiquement des déformations réalistes pour renforcer les applications d'évaluations de conceptions industrielles.

Deuxièmement, nous examinons la déformation physique à base de méthodes de modélisation dans le contexte de l'interaction haptique. Il est vrai qu'une grande quantité de travail a été faite sur la simulation interactive basée sur les modèles physiques des objets élastiques (par exemple dans l'infographie). Toutefois, seul un sous-ensemble relativement petit de ces travaux ont déjà abordé les besoins de simulation du retour haptique. Deux catégories principales de solutions peuvent être trouvées pour résoudre le défi: soit on applique des modèles simplifiés basés sur la linéarisation ou la réduction,

on effectue des pré-calculs (hors ligne) avant d'entrer dans la phase d'interaction en temps-réel. D'une part, nous présentons les travaux, qui visent à améliorer la précision des modèles ou qui visent à l'accélération de la vitesse de calcul des modèles éléments finis. D'autre part, la méthode de réduction de modèles est couramment appliquée pour projeter l'espace de déformation initiale de l'état sur un sous-espace de plus faible dimension pour parvenir à un système de taille réduite mais en conservant l'essentiel de sa capacité à représenter les propriétés mécaniques du système d'origine. Le choix de l'espace de déformation pour représenter le comportement en temps-réel d'objets déformables est de première importance. Les méthodes basées sur les données et la méthode d'analyse modale partagent le même but de calcul des espaces de représentation des déformations dans une phase hors-ligne.

La méthode de réduction de modèle est attrayante pour l'interaction haptique en temps-réel avec des simulations de déformation, car la plupart du temps de calcul pour la modélisation et la simulation de ces objets déformables peut être effectuée au cours du processus de pré-calcul. Par conséquent, la puissance de calcul au cours de la phase d'interactions en temps-réel peut alors être consacrée à d'autres modèles de comportement déformable, comme la visco-élasticité, et la non-linéarité. Cependant, la technique de réduction de modèles basée sur des pré-calculs soulève plusieurs difficultés en termes de compression des données recueillies, de conception d'une stratégie efficace de collecte des données et d'adéquation des données pré-calculées pour différents scénarios. En revanche, la méthode en deux étapes proposée dans cette thèse permet de contourner efficacement ces difficultés susmentionnées qui sont soulevées par la méthode de réduction de modèles.

Chapitre 4

Dans ce chapitre, nous présentons une description détaillée de notre modèle de déformation haptique, qui est un modèle de calcul reliant la modélisation de la déformation, d'une part, et les interfaces haptiques d'autre part. L'objectif principal de ce chapitre est de décrire les flux de données de l'interaction haptique avec des objets déformables. En particulier, nous avons discuté le modèle haptique à partir de trois aspects: les éléments clés de la modélisation par éléments finis, la modélisation du mouvement des corps rigides si nécessaire, les interfaces haptiques en mode admittance et le flux de données associé.

En premier lieu, nous présentons un aperçu de notre modèle haptique en termes de modélisation par éléments finis, d'interface de contrôle en mode admittance et de flux de données dans le modèle. (1) Les données du maillage volumique sont appliquées pour la procédure d'assemblage des matrices EF, tandis que les données de maillage de surface sont utilisées principalement pour la définition des conditions aux limites. Lorsque l'on utilise la FEM, les conditions aux limites (CL) sont décrites par les valeurs prescrites des variables (par exemple, le déplacement) sur certaines parties de la surface du corps déformable. La modélisation du comportement d'un objet déformable et mobile peut être calculée en résolvant les équations classiques. (2) Après avoir précisé les principales composantes de la modélisation de la déformation, nous portons l'accent sur l'interface haptique. Le mode de contrôle en admittance des dispositifs haptique mesure la force appliquée par l'utilisateur et la contrainte sur la position en déplaçant l'appareil. Par conséquent, la sortie fournie par l'interface haptique est représentée par un vecteur de données de forces qui contient les composantes x, y et z des forces ou des couples. De

manière symétrique, la seule entrée qui est délivrée à l'appareil est représentée par un vecteur de données qui contient la position x, y et z d'une sonde virtuelle représentant l'extrémité du dispositif dans l'espace 3D et, pour le cas de dispositifs 6-DDL, les angles de roulis, de tangage et de lacet. (3) La liaison entre les composants de la modélisation par éléments finis et l'interface haptique est établie pour définir la boucle principale du modèle haptique. Les données en effort sont précisées de sorte que la sortie de l'interface haptique puisse être associée aux conditions aux limites (CL) et sont considérées comme l'entrée de la modélisation par éléments finis. De même, les données de position de sortie de la sonde virtuelle calculées par le modèle en éléments finis peuvent être identifiées et transmises à l'interface haptique.

Deuxièmement, nous nous concentrons sur la méthode d'analyse de maillage basée sur des scénarios prévus et définis dans le cadre de notre application industrielle. Au cours des pré-calculs hors-ligne, il n'est pas nécessaire de mailler l'objet par l'application d'une résolution de maillage unique sur l'ensemble de l'objet complexe déformable. En effet, lorsqu'un utilisateur interagit avec des nervures ou des raidisseurs locaux à des fins d'évaluations de conception, l'interaction et la déformation se concentrent dans une région locale. Sur ce point, nous proposons une méthode d'analyse de maillage local dédiée à ces scénarios de vérification de déformation. Cette méthode peut être considérée comme une tentative vers l'amélioration de la précision de la déformation par le découpage de l'objet déformable en volumes d'intérêt (VI) distincts. La plupart des temps de calcul des préparatifs des maillages a lieu hors-ligne, de sorte que les déformations du VI, dans les scénarios prévus, peuvent être calculés sur la base de maillages plus denses, tout en respectant l'objectif de réduire la charge de calcul en ligne grâce à un système de basculement entre ces différents espaces de déformation modale pré-calculés.

Troisièmement, pour les interactions en temps réel haptique, nous appliquons un modèle d'interaction ponctuelle. Contrairement à leurs homologues rigides, un soin particulier doit être pris avec des modèles élastiques pour définir les domaines finis de contacts pour les interactions ponctuelles. En effet, des points de contact définis comme un seul sommet ou un point plus proche voisin sur le maillage conduisent à des artéfacts de maillage dans la simulation. Cependant, nous ne considérerons pas la définition de "faible surface de contact" au cours des interactions haptiques en temps-réel dans la thèse. Le terme "point unique" signifie un sommet de surface unique qui est associé à la représentation virtuelle de l'effecteur du dispositif haptique. De même, l'utilisateur peut manipuler un objet virtuel déformable utilisant les deux mains, ce qui correspond au scénario d'évaluation de la conception globale présentée dans la section 1.4.2. En outre, un tel paradigme est utilisable pour les utilisateurs qui interagissent conjointement avec le même objet déformable et qui sentent l'influence de l'interaction due à l'autre utilisateur.

Chapitre 5

Comme nous l'avons vu dans les chapitres précédents, la base physique pour l'interaction haptique avec des objets déformables formulée par la FEM et la théorie de l'élasticité conduisent à un système énorme et, de ce fait, le processus de résolution est coûteux ce qui empêche l'interactivité en temps réel. L'une des principales contributions de la thèse est de proposer une méthode de modélisation de la déformation en deux étapes à des fins d'interaction haptique en combinant une phase de pré-calcul hors-line et une

phase d'interaction en ligne. En utilisant cette méthode, une partie importante des tâches de calcul est effectuée hors ligne et, par conséquent, nous obtenons un modèle de déformation à coût calcul faible qui assure une expérience d'interaction stable en temps-réel.

Le pré-calcul est largement employé dans les cas de simulations interactives de déformation pour alléger la charge de calcul en temps-réel, et différentes stratégies sont conçues à des fins de pré-calcul différentes. Dans cette thèse, la phase de pré-calcul est dédiée à un flux de données visant à l'obtention d'espaces de déformation dans lesquels les déformations en temps-réel peuvent être approchées pour traiter les compromis entre l'exactitude de la déformation et la performance de l'interaction haptique. Comme notre phase hors-ligne repose sur des logiciels différents, la transformation du format des données est indispensable. Une amélioration possible serait d'intégrer l'ensemble du processus de pré-calcul dans une seule interface. Le flux de données principal des pré-calculs hors ligne part de la modélisation géométrique dans un logiciel de CAO (par exemple, CATIA™), du modèle de maillage obtenu dans GMSH, et utilise le Code-Aster pour le calcul d'analyse modale.

Pendant la phase d'interaction haptique en temps-réel, la tâche principale est de calculer la réponse de l'objet déformable en fonction de deux aspects: la réalisation de la boucle de rendu haptique et le calcul de déformation de l'objet entier. Dans cette thèse, la réalisation de la boucle de rendu haptique dépend du contrôle en mode admittance des dispositifs haptiques et des étapes d'interaction en temps réel présentées dans la section 4.3.2. Les blocs clé de notre phase en temps-réel sont représentés dans la partie droite de la Figure 5.1, de la sensation des forces de l'utilisateur, au calcul des forces modales, puis au calcul du déplacement réduit et du déplacement physique et finalement à la position finale et la réaction de l'interface haptique. Plus particulièrement, nous introduisons un système en ligne de division de la déformation pour répondre, d'une part au taux de rafraîchissement élévé de la boucle de rendu haptique, et, d'autre part, au taux de mise à jour de la boucle de rendu graphique. Le noyau de ce système découple la boucle haptique de la boucle graphique. Elle sont mises en œuvre sur différents processus. (1) Dans le processus de rendu haptique, quand le point d'interaction (HIP) prend le contrôle d'un sommet actif, basé sur les étapes d'interaction, l'indice (par exemple,le i^{eme} vertex) du sommet actif est enregistré, le sous-espace modal correspondant est choisi par l'interrupteur de choix de sous-espaces puis une sous-matrice $\Phi^i = \{\varphi_1^i, \varphi_2^i, ..., \varphi_r^i\}$ est extraite de la matrice modale pré-calculée Φ (voir la Figure 5.6 montrant l'extraction); les forces appliquées à l'effecteur sont alors enregistrées dans $\mathbf{F}$ et la force modale correspondant $\mathbf{f}$ est calculée par l'équation 2.27; les réponses de chaque mode en termes de déplacement résultant et sont calculées par l'équation 2.29 sont assemblées dans $\mathbf{q}$. (2) Dans la boucle de visualisation, la déformation de l'objet entier est calculée en fonction de l'équation 2.20 et de l'équation 2.19 en utilisant l'histoire des vecteurs de déplacement réduits $\mathbf{q}$, qui est transmise par le processus de rendu haptique par l'intermédiaire d'une mémoire partagée. Ensuite, les déformations sont rendues avec une baisse du taux de rafraîchissement dont la valeur dépend fortement de la densité de maillage.

Comme nous considérons le mouvement de corps rigide d'un objet déformable, notre modèle haptique doit être adapté à ce cas. La Figure 5.7 illustre l'idée clé de la séparation et de l'intégration, par application du théorème de superposition, du mouvement de corps rigide globale et de la déformation linéaire. En particulier, les modèles de

calcul de mouvement de corps rigide et de déformation sont définis séparément sur la base de la méthode en deux étapes proposée dans cette thèse. Dans notre phase de pré-calcul hors-ligne, le comportement global qui est calculé par la simulation par éléments finis est séparé en une composante du mouvement de corps rigide et une composante de la déformation; La composante du mouvement de corps rigide est d'abord éliminée et la composante de déformation qui est représentée par l'espace modal de déformation est enregistrée. Au cours de la phase d'interaction en ligne avec la déformation, le mouvement de corps rigide est intégré en utilisant les équations classiques du mouvement (Principe fondamental de la dynamique), tandis que la déformation est calculée en fonction de notre méthode de calcul de déformation en deux étapes. Le comportement global est obtenu en combinant (superposant) les composantes du mouvement de corps rigide et de la déformation.

Chapitre 6

Dans ce chapitre, nous décrivons la mise en œuvre de notre système de simulation de déformation en temps réel par l'introduction de la configuration matérielle, des données statistiques de nos modèles expérimentaux et du développement de logiciel en termes de fonctionnalités principales. Ensuite, nous discutons les résultats expérimentaux à partir de deux aspects, que sont l'exactitude de la déformation calculée et la performance de l'interaction, en fonction des différents modèles et du choix de différents nombres de modes. En ce qui concerne les interactions en temps-réel, nous présentons un scénario d'interaction haptique avec des objets fixes déformables dans le but d'illustrer les scénarios locaux de vérification de déformation. Le second scénario d'interaction haptique, avec des objets mobiles et déformables, correspond au scénario global de vérification. Enfin, nous présentons notre banc expérimental et les premiers résultats obtenus dans le cadre de la coopération avec PSA.

Nous proposons une conclusion en indiquant l'importance et les points forts du travail présenté. D'une part, il existe plusieurs applications industrielles potentielles qui pourraient bénéficier de la modélisation de la déformation et de la méthode d'interaction en temps-réel proposée. Un premier exemple est la simulation de déformation en temps-réel pour l'intégration de l'analyse interactive et de re-conception dans un environnement de réalité virtuelle. Un autre exemple est l'introduction de la prise en compte des déformations volumétriques dans la procédure d'assemblages virtuels ou dans celle des opérations de maintenance guidées par les interfaces haptiques. D'autre part, les travaux futurs en vue de l'amélioration et l'extension de notre méthode seront principalement axés sur les trois aspects suivants. (1) Pour répondre au faible taux d'interactivité de certains modèles CAO industriels trop complexes, une solution pourrait être de simplifier la géométrie avant le pré-calcul hors-ligne, car la simplification géométrique fait partie du processus d'analyse par éléments finis. (2) Nous limitons les possibilités d'interaction à une sélection de manipulations possibles, ce qui signifie que le processus de pré-calcul doit être ré-effectué pour chaque nouveau scénario d'interaction non prévu. Notre méthode en deux étapes dépend fortement de la description initiale des conditions aux limites. C'est un travail important à l'avenir d'étendre la capacité de nos méthode pour permettre des interactions arbitraires en prenant en compte des conditions aux limites variant dans le temps ou par la génération de "pré-calculs" associés au cours même de la séquence d'interaction en utilisant des ressources de calcul distribué. (3) La visualisation de la déformation pourrait être étendue pour

l'industrie mécanique. Il sera sans doute nécessaire d'accéder à d'autres types de résultats comme les champs de déformation et les champs de contraintes. Ceci pourrait aider les concepteurs à mieux comprendre le fonctionnement global de l'objet dès la phase de conception.

Contents

List of Figures

List of Tables

General Introduction

The advent of computer technology has revolutionized every aspect of daily life or professional ones. And more particularly, the information processing has been benefiting from such tremendous progress. In this thesis, we concern a more precise case: the Product Lifecycle Management (PLM), which is treated by more rigorous information technologies. Thus, the PLM provides an abundant of tools to define products, combines all the information related to the products, and exchanges such information between different actors in their lifecycles. Among these tools, Computer Aided Engineering (CAE) has the advantages to create and edit a product in a digital format. And moreover, it supplies numerical simulation methods to analysis the functionalities of a product. The primary advantage of a digitization process is the reduction of time and costs in terms of the product modifications and the product data exchange in the PLM.

Concerning the evaluation of a product functionality, some CAE simulation tools are available. However, the main drawback of such simulations lies in two aspects. On one hand, the implementation of a CAE simulation, e.g., Finite Element Analysis (FEA), tends to be a time-consuming process to achieve a desirable analysis result, especially for the cases in which a complex industrial mechanical part is considered. On the other hand, the interaction modes during these simulations remain relatively poor. For example, engineers normally can not access to the intermediary simulation results to adjust simulation parameters in an interactive way. These disadvantages of current CAE simulations decrease the efficiency of information flow and consequently, incline to influence the process of the PLM in a negative way.

In another area, the development of information technology boosts the invention of new solutions with advanced technical equipments which bring the human operators closer to the scientific data from different fields. For example, the Virtual Reality (VR) is a promising domain in which an operator is immersed in the product space characterised by realistic renderings, multi-sensories, and intuitive interactions. Thus, the VR technology unseals a terrace with a large variety of potential applications, ranging from massive scientific data explorations, surgical trainings, to virtual prototypings.

Regarding the virtual prototyping in a PLM, the work presented in this thesis is within the framework of the industrial evaluation of deformable mock-ups in a VR environment by introducing human operators into the loop. Based on the industrial practices, an interactive design validation of such mechanical parts plays an important role in a PLM, because an interactive deformation simulation in a VR environment allows engineers from different industrial sectors to "step into" the screen to manipulate these digital mock-ups for the purpose of identifying design problems (e.g., the insufficiency of a structure) in the early design phase. In this way, the time and costs required for sharing product information among different sectors would be largely reduced, and therefore the efficiency of the design information exchange in a PLM can

be considerably increased.

However, an interactive deformation simulation in a VR environment through haptic interfaces is a challenging task caused by the trade-off issue between the deformation accuracy and the real-time interaction performance. This issue would be troublesome for simulation cases in which complex mechanical deformable parts are taken into account, because the degree of deformation realism tends to require a high meshing resolution, which in turn, has a direct impact on the size of the matrices involved in the elasticity system and thus brings about a huge computational task in real-time. In this thesis, we propose a two-stage deformation simulation method for real-time haptic interaction purpose, by combining an off-line pre-computation phase and an on-line deformation interaction phase. Experimental results show that our method can efficiently handle the trade-off issue, because the deformation modelling is formulated by the finite element method which guarantees the deformation accuracy; and moreover, the heavy computations of large elastic systems are occurred off-line which assure a costless deformation response model in real-time.

Thesis Organization

The structure of this thesis is organized as follows.

The first chapter describes the context of the thesis: the foundations of the PLM and of the VR. First, we present an overview concerning the applications of the VR technology towards the PLM. Second, various applications developed in the VR environment are discussed and we focus on industrial applications which aim at the integration of the VR and the PLM. Finally, we describe the motivations of the thesis: to improve the interaction methods in traditional CAE applications, with an objective to reduce the time and costs needed for an industrial PLM by interactive deformation evaluations concerning deformable mechanical parts.

The second chapter explains the necessary theoretical backgrounds for the deformable objects modelling in the thesis. It is composed from the following parts. First, a brief survey on deformation modelling methods are presented, and we emphasize that the physically-based deformation modelling methods are suitable for industrial design validation of deformable parts through haptic interfaces. Second, we describe the theory of elasticity from which the physically-based modelling methods are derived. Third, the finite element method (FEM) and the equation of motion are discussed and we point out the computational challenges of employing FEM for real-time deformation simulations. And then we introduce the basic formulations of the model reduction method as well as the modal analysis method. At last, an overview of finite element software or solvers used for the deformation modelling and analysis is given, as these software are commonly applied for off-line pre-computations.

The third chapter presents a state-of-the-art concerning two aspects. First, different research fields of haptics employment are clarified and we emphasize that the computer haptics is our research interest, as it contributes to haptic rendering of realistic deformations and thus guides and strengthens the applications of industrial design

evaluations. Second, we present various existing methods aiming to handle the trade-off issue of haptic interaction with physically-based deformable objects.

The fourth chapter describes the haptic deformation model employed in the thesis. The main role of haptic deformation model is to represent the data flow between the deformation modelling part on one side and the haptic interaction part on the other. We present our haptic deformation model from three aspects. First, we explain an overview of our haptic model in terms of the main components of FE modelling, the admittance-controlled haptic interfaces and the data flow in the haptic model. Second, a description of modal deformation space is presented as we apply modal model to represent real-time deformations. Regarding the off-line construction of a deformation space, we introduce, in details, the mesh analysis method based on anticipated design evaluation scenarios. And moreover, the real-time switch scheme among different modal deformation spaces is presented to allow a real-time switch (transition) among different modal spaces with respect to the interaction contexts. Third, we introduce the point-like interaction model and focus on our interaction steps based on the admittance control of haptic interfaces and single point interaction model.

The fifth chapter presents our two-stage method by describing the off-line precomputation phase and the on-line deformation interaction phase separately. Concerning the off-line phase, we described the data flow from the modal preparation, to the meshing procedure, and finally to the computation of modal deformation space. Regarding the on-line phase, we focused on the deformation division scheme which was dedicated to solve the coupling issue of high refresh rate haptic interaction with relatively slow deformation simulation. At last, an experimental bench was briefly presented by illustrating an operator's different interaction operations with a deformable mechanical part.

The sixth chapter presents the implementations of our proposed method in the thesis and the experimental results. First, we introduce the hardware configuration and the statistical data of our experimental models. Second, the software development for the deformation evaluation framework is described in terms of the main functionalities. Third, we discuss the experimental results from two aspects regarding the different models and the different number of modes: the deformation accuracy and the real-time interaction performance.

And finally, we finish the thesis by concluding our work and presenting several perspectives in the conclusion.

Virtual Reality and Product Lifecycle Management

1

Contents

The first studies on virtual reality began some fifteen years ago in the USA. One example was that the CAVE™ was invented at the University of Illinois[1]. To start with, virtual reality was employed primarily in research fields and its scope did not extend beyond the scientific sphere. Then, industries started to take an interest. Today, virtual reality is applied in an increasing number of industrial sectors: the aerospace industry for cabin layout and maintenance; surgery, for the teaching of techniques; as well as in energy, railways, ship-building; and of course, the automotive industry, which is one of the key customers of virtual reality. Indeed, virtual reality provides first-person interactions in real-time to the studied virtual prototypes, and thus the simulation results on these mock-ups can be transmitted efficiently among different industrial sectors in a Product Lifecycle Management (PLM). Therefore, engineers can forecast the design or usability problems of the virtual prototypes during the first steps of the design process. And moreover, the employment of virtual reality technologies in the industrial background increases efficiency of sharing design information in a PLM

[1] www.illinois.edu

and thus contributes, to a large extent, to the production of innovative products to maximize customer interest and sales.

The work presented in this thesis is within the framework of employing the virtual reality technology for the real-time deformation evaluations which are key components towards the PLM concerning industrial deformable mechanical parts. In this chapter, we present an overview of the technology of the PLM and the Virtual Reality (VR), focusing on recent industrial applications which aim at the integration of the two domains toward the design evaluations in a early design phase.

1.1 Introduction

The work in this thesis can be viewed as an intersection between two domains: the PLM and the VR. First, we present the important role of the PLM in the industrial background, its evolution and the advantages of digital representations of a product. With the enormous development of information technology, the PLM integrates a variety of Computer Aided Engineering (CAE) tools, which have undergone considerable changes by providing more efficient design solutions. However, these tools manifest drawbacks, e.g., the time-consuming computations of performing an analysis, and the poor interaction methods during the computation process. Actually, the VR is a promising area which aims at providing first-person manipulations with interactive, intuitive and immersive experience between an operator and a digital mock-up in a virtual environment. It is for this reason that the VR applications can largely promote the crucial role of the PLM in automotive industries. Industrial applications presented in the subsequent sections equally prove the advances of the PLM due to the employment of the VR. Regarding our research, it is the possibility of experiencing the deformation of mechanical parts through haptic interfaces.

1.2 Product Lifecycle Management

1.2.1 Overview

The PLM of a product concept is a strong presence in automotive industries, which can manage a product by exchanging information relating to all stages of its existence. This information is exchanged between different actors in the whole process in the PLM. Better tracking of a product in the company provides a greater continuity and conformity between the conception design office and the production line. Furthermore, the competitiveness of world market urges automotive companies to be reactive to diversify their new products. In such a background, the engineers are forced to reduce the time and costs needed for a product construction and evaluation [Bullinger et al., 1999].

The PLM can also serve for marketing, ranging from services for planing advertisements to services for purchasing and logistics, so that necessary design requirements towards a new product can be predicted and adopted.

Finally, after-sale services will obtain useful comments about a product from consumers. The PLM induces a better traceability and reduces production time due to an efficient exchange of design information among different sectors. It also improves the

management of resources and allows the reuse of certain design knowledge for future products.

1.2.2 Product Lifecycle Management Phases

The management of a product can cover all steps of a production or a part of it. The process of realization of a new product can be decompose into several steps, from a new concept proposition to the after-sale services. We refer to the work of [Picon, 2010] to detach a PLM by four phases (see Fig. 1.1): a conception design phase, a CAD model design and evaluation phase, a manufacturing phase, and an operation phase. However, in this thesis, we are interested in the product design evaluation process of industrial mechanical parts (see the dot dashed rectangle in Fig. 1.1). Comparing with traditional numerical simulations, we propose a real-time deformation simulation method which enables us to experience real-time deformations interactively.

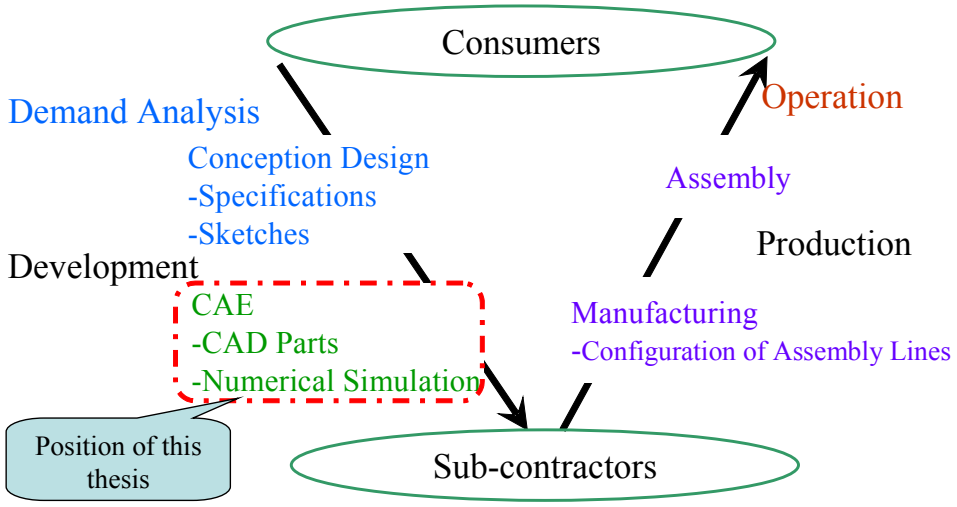


Figure 1.1 – A general representation of the production cycle of a product. It begins with the expression of needs and ends with the operation. The design steps are iterative and editing CAD until obtain a satisfactory model. Simulations are necessary to check before manufacturing a physical prototype the product for the assembly lines (inspired by [Picon, 2010]).

1.2.2.1 Conception Design

The first stage of a PLM is to consider different constraints and properties of a new product based on the market demands and consumers' requirements. During this step, the conception design is usually based on the employment of drawings with paper and pencil (sketches). This phase allows a conception designer to imagine freely, while respecting equally the specifications of the product. It should be emphasized that this phase is an iterative process, because necessary modifications will be implemented with respect to the feedback from digital or physical models of the concept.

1.2.2.2 CAD Parts Design and Validation

During this phase, CAE tools (e.g., CATIA™) or laser scanning equipments are usually applied to achieve a precise digital model of the product. This digital model (or termed

"a virtual prototype") possesses many advantages comparing with the corresponding physical one: it allows an efficient data exchange between the various stages of the development of a product, and it is commonly employed to perform numerical simulations. These simulation tools provide possibilities to implement extensive evaluations for checking the functionalities of a product. In our research, we focus on industrial mechanical parts in the context of the real-time deformation validation.

1.2.2.3 Manufacturing and Product Assemblies

The manufacturing of a product includes various stages of production, ranging tools development to procedures for the completion of a product. Before physical models are manufactured, the digital mock-ups are employed to perform tests of assembly/disassembling, designing assembly lines, simulating the manufacturing product. The model will also serve to test the ergonomics of the worker (fatigue) on this string, or ergonomics of the product. Finally, a check of the conformity between the numerical model and the product manufactured is necessary.

1.2.2.4 Operations and Services

In this stage, the product is finished and available for sale on the market, followed by the management of after-sale services ranging from technical supports, reparation, to maintenance procedures. It is also in this stage that users' feedback concerning the degree of satisfaction are analysed.

It is necessary to point out that if more than a modification comes late in the cycle of a product, plus associated costs (financial and delays) will arrive for the company. The correct decisions taken during the early design phase play therefore a very crucial strategic role during the whole development of a product [Kardi, 2007].

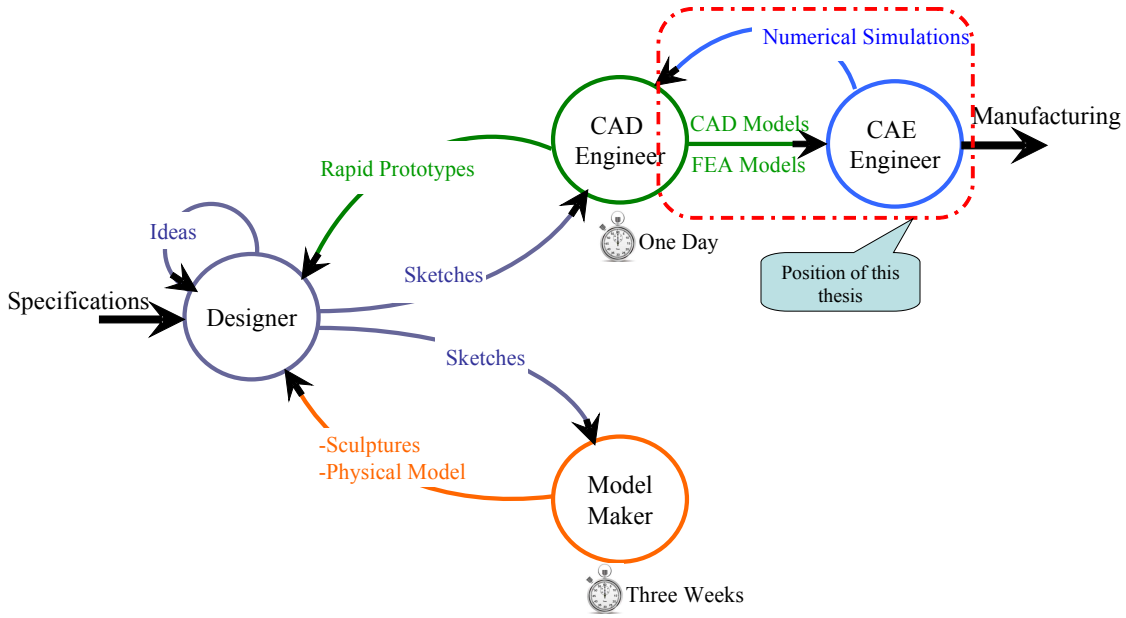


Figure 1.2 – Representation of the four actors that can be found in a product design cycle. It should be noted that the process is evolutionary and that several iterations are necessary between different actors. The cycle of digital update is faster that the creation of a physical model. Inspired by [Picon, 2010].

1.2.3 Virtual Prototype versus Physical Prototype

Virtual prototyping is becoming a commonly adopted design and validation practice in several industrial sectors [Dai et al., 1996; Zorriassatine et al., 2003]. Companies are moving from expensive physical prototypes to the direct realization of digital mock-ups. Compared to physical prototypes, virtual prototypes manifest several advantages. And among them we intend to emphasize the following points:

- the reduction of development time and the reduction of the costs physical prototype;
- making the evolution of model facilitative in terms of keeping the model up to date with respect to the conception design;
- the possibility of applying a digital model to create documentation or as a link between the different actors in the process (see Fig. 1.2);
- the possibility of implementing collaborative work in a remote distance in terms of the data sharing and simultaneous work on a same model;
- for reviewing the rapid change of aesthetic parameters of the object (color, texture, etc.);
- the verification of the possibility of assembling different objects, or the verification of compliance/stability.

Physical prototypes are generally replaced by virtual prototypes due to their advantages discussed above. As we will present in section 1.4.2, this thesis is based on the employment of virtual prototypes of a physical stamping mould for the purpose of design evaluations.

Nevertheless, it should be pointed out that physical models are still be utilized, as there exist certain evaluation cases where numerical models can not be competent. For example, as shown in Fig. 1.3, designers are checking a clay model of a car body[2] by manual operations to judge whether all requirements are fulfilled in terms of the target vision or the surface of the modelled sculpture.

Figure 1.3 – Designer checking the physical form of a sculpture clay model. Images from BMW.

[2] http://www.bmwblog.com/2008/08/08/behind-the-design-of-the-bmw-7-series/

1.2.4 Problems of Current CAE in Product Lifecycle Management

As illustrated in Fig. 1.2, numerical simulations have to be carried out before entering into the manufacturing process. The primary tasks for CAE engineers is to implement a serial of validations with respect to the design requirements or the constraints from real working conditions. Not only CAD engineers but also conception designers benefit from such simulation results. There are two reasons why engineers are obliged to reduced the time needed for evaluating a product. The competitiveness of world markets is an external cause pressing them to diversify their products and to release new products. An internal cause is that the PLM requires an efficient and expeditious way to exchange design knowledge of a product during the whole lifecycle.

Traditional numerical simulations are based on several CAE simulation tools. However, the main drawback of such simulations lies in two aspects. On one hand, the implementation of a CAE simulation, e.g., Finite Element Analysis (FEA), tends to be a time-consuming process to achieve a desirable analysis result, especially for the cases where a complex industrial mechanical part is considered. On the other hand, the methods of interaction during these simulations remain relatively poor. For example, engineers normally can not access to the intermediary simulation results to adjust simulation parameters in an interactive way. These disadvantages of current CAE simulations incline to influence the process of the PLM in a negative way, as the efficiency of information workflow decrease.

By contrast, the VR is an area in which one seeks to immerse an operator in the product space by offering an environment featured by an immersive rendering and an intuitive ways for interactions.

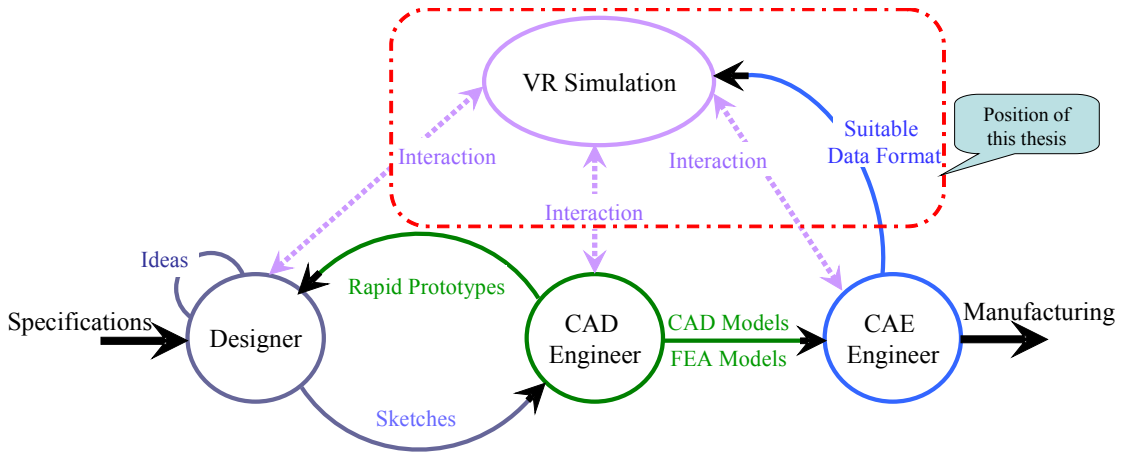


Figure 1.4 – Representation of the design cycle with the introduction of VR. It will play an important role in the interaction between the conception designer/CAD engineer/CAE engineers and the digital model. Currently the VR tools are employed to encourage exploration of the model and some simulations (assemblies, usability testing, etc.). Note that the use of VR requires a change of format data. Inspired by [Picon, 2010].

Figure 1.4 illustrates a representation of the design cycle with the introduction of VR. The VR simulation module provides interactions between conception designers/AD engineers CAE engineers(designer, model maker, CAD engineer, CAE engineer[3] and the digital models. In this thesis, the VR simulation module represents a

[3]Mechanical CAD and CAE are the domains with two different groups of engineers, with different toolsets.

real-time deformation evaluation framework aiming at the design validations of deformable mechanical parts.

In the following sections, we present the foundations of the VR. Specially, we introduce various applications to demonstrate the potential of the VR and we focus on the existing VR-PLM integration applications in the industrial background.

1.3 Foundations of Virtual Reality

Virtual Reality (VR) becomes a substantial and ubiquitous technology and subsequently penetrates into the fields of education, learning and training. In a VR environment, computer graphics is combined with advanced display systems and control devices to provide an immersive and natural interaction with the objects in the environment. In this way, natural interactions with the objects in the real world can be applied in various simulations, and real situations can be examined in a fully controlled virtual environment. For example, building a digital model for evaluating its functionality for medical educations, or for virtual prototyping applications. More promising applications for learning purposes is to let the user manipulate objects and experience the consequences [Burdea, 2003]. As explained in [Schuemie et al., 2001], the factors that distinguish VR from other technological systems are the characteristics in terms of *presence* and *interaction*. The sense of presence gives a user the illusion of being physically present in a VR environment, while the sense of interaction provides the chance to manipulate and modify the objects within the environment.

In the following section, we present a brief overview of different applications developed in VR environment. And we focus on industrial applications which aim at integrating the VR technology with the PLM.

1.3.1 Virtual Reality Applications in General

Scientists and engineers dedicate efforts to the progressive development of applying the VR technology in a variety of domains over the last fifty years. However, most of the technological advances have been made in the last ten years, undoubtedly due to improvements in computer efficiency and the miniaturization of sensorization devices. And thus, a large majority of applications in different fields have emerged. In the following paragraphs, we summarize several typical VR applications to demonstrate its potential and we claim that the VR technology can be equally employed towards the PLM for the purpose of industrial deformation validations in the product early design phase.

Tele-presence and tele-operation. VR contributes to build systems that enhance communication by facilitating information display. In tele-operation applications, VR improves the performance of accomplishing a task due to the intuitive manipulation and exploration of a working environment. In such environment, an interactive 3D simulation provides a "virtual assistance" to the operator by adding guides or actors, which aid the operator in task learning or execution [Fong et al., 2000; Rodriguez et al., 2002].

Data visualization and exploration. Data modelling is an attractive method to represent massive, multi-variates, and time-varying 3D data. To gain an insight into such

complex data by overcoming visualization and exploration challenges, several techniques have been developed and employed within VR environments, and these techniques provide the user sufficient control over the visualization interactively [Lin et al., 2008; Griffith, 2010].

Virtual sculpture. Virtual sculpture is a modelling technique for computer graphics based on the notion of sculpting a solid material with tools in reality. In the process of product conception design, not only designers with professional CAD skills, but also final customers participate in the design and evaluation of product shape styling. Some methods are proposed to supports customers manipulation of sculpting constraints in VR environments by controlling the deformation of product shape in an intuitive way [Liu and Tan, 2002].

Surgical training. Surgical training has traditionally been one of apprenticeships, where a surgical trainee learns to perform the surgery under the supervision of a trained surgeon. However, this is costly and time consuming. Training systems which employs VR-based simulator are alternatives to supplement standard trainings, and such systems are gaining popular recognitions [Jerabkova et al., 2005]. Recently, several multimodal frameworks which provide a rich visual, auditory, and touch-enhanced virtual surgical world have enabled the transfer of VR-based trainings to actual skills [Weiss et al., 2003; Indhumathi et al., 2009].

Augmented reality. Concerning a VR environment, a user cannot observe the real world around him once immersed. Augmented Reality (AR) is a variation of VR, as an AR environment allows a user to observe the real world, with virtual objects superimposed upon or composited with the real objects [van Krevelen and Poelman, 2010]. Ideally, AR would appear to the user that the virtual and real objects coexisted in the same space rather than completely replacing it. Since AR is a complement for VR, applications in AR can be ranged from medical trainings [Vogt et al., 2006], large-scale military scenarios [Julier et al., 2000], engineering educations [Liarokapis et al., 2004], to gaming systems [Piekarski and Thomas, 2002], etc.

1.3.2 VR-PLM Integration Applications

As described above, the VR technology has been applied in different fields due to its intuitive, interactive and immersive features and in turn, these applications have benefited a lot from the employment of the VR technology. Following on from these successful applications, a question rises of whether we are ready to apply it to industrial applications with an objective of integrating with the PLM. Actually, companies like Boeing [Schmitz, 1995], Volkswagen [Dai et al., 1996], Chrysler [Mahoney, 1997], Ford [Deitz, 1995], Caterpillar [Mahoney, 1995], General Motors [Kobe, 1995] have been evaluating the employment of the VR technology to reduce the number of physical prototypes by simulations implemented on digital mock-ups in a virtual environment. It is necessary to point out that PSA Peugeot Citroën[4], one of the most advanced industrial user in the world of VR and especially in the field of haptics for assembly

[4]http://www.psa-peugeot-citroen.com

simulations, has been carrying out in-depth evaluations of the employment of VR to reduce the cost of physical prototypes and to generalise its utilisation in the industrial conception process.

In the following subsections, we classify industrial VR-PLM integration applications by four different themes: VR for assembly/disassembly applications, VR for projects review, VR for ergonomics analysis and VR for interactive deformation simulations.

1.3.2.1 VR for Assembly/Disassembly Applications

Recently, the VR technology together with a variety of human-computer interfaces empowers industrial evaluation applications with a faster and more powerful quantitative performance analysis. Among these industrial applications, the most dominant ones are virtual assembly simulations. It has been established that assembly processes often constitute the majority of the cost of a product [Boothroyd and Dewhurst, 1989]. Thus, it is crucial to develop a proper assembly plan early in the design stage. A well-designed assembly process can improve production efficiency and product quality, reduce cost and shorten product's time to market. For example, a knowledge-based virtual constructor system based on natural language commands was developed by [Jung et al., 1998] and the system enabled an interactive assembly of 3D mechanical parts. Regarding the disassembly and maintenance processes, authors of [Sung and Corney, 2001] described a system which could automatically recognize assembly features and generated disassembly sequences.

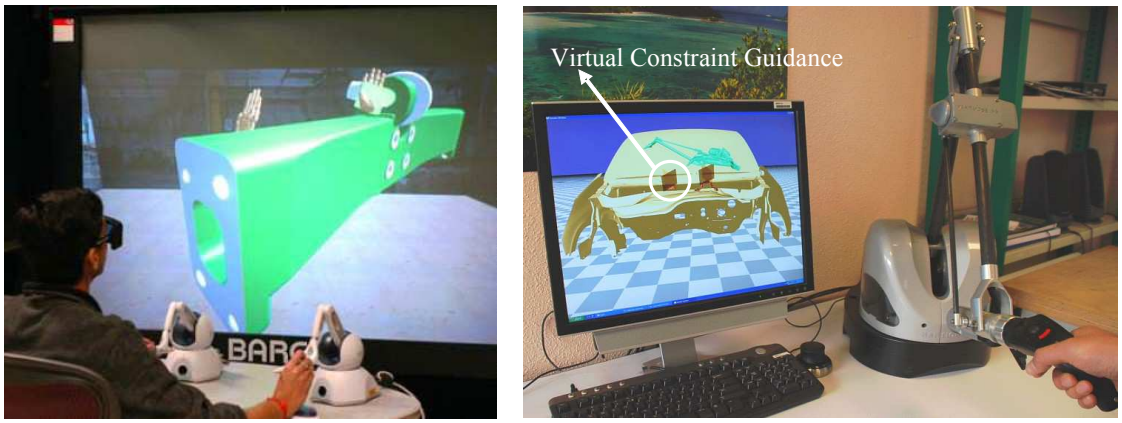


Figure 1.5 – Two typical haptic assembly applications concerning rigid-body dynamics. The left picture is from [Seth et al., 2006], while the right one is from [Tching et al., 2010].

It is necessary to point out that modern CAD systems are also used in assembly process planning by manually selecting the mating surfaces, axes and/or edges to assemble the parts. However, these interfaces do not reflect human interaction with complex parts, as such computer-based systems lack in addressing issues related to ergonomics, such as awkward to reach assembly operations. By contrast, applying haptics technology, engineers can touch and feel complex CAD models of parts and interact with them using natural and intuitive human motions [Volkov and Vance, 2001]. Collision and contact forces calculated in real-time can be transmitted to the operator using robotic devices making it possible for him/her to feel the simulated physical contacts that occur during assembly operations. For example, authors of [Seth et al., 2006] developed a System for Haptic Assembly and Realistic Prototyping (SHARP)

which employed physically-based modelling for simulating realistic part-to-part and hand-to-part interactions through a dual handed haptic interface (see the left picture of Fig. 1.5). To ensure a good assembly of CAD objects, authors of [Tching et al., 2010] proposed a method for interactive assembly operations by applying both kinematic constraints and guiding virtual fixtures (see the right picture of Fig. 1.5). From an assessment point of view, a quantitative analysis concerning the significance of haptic interactions in performing simulations of manual assemblies was performed by [Vo et al., 2009].

1.3.2.2 VR for Projects Review

The project review is a crucial step of evaluating a digital mock-up towards its production. Engineers apply mostly large devices to view the rendered stereoscopic. And it is during this stage that some tests can be carried out, e.g., simulations and verifications of the general appearance of the product (color, texture) in different lighting conditions.

(a) Exterior appearance design review (b) Internal design review

Figure 1.6 – (a) PSA's digital imaging room; and (b) Ford's product visualization center.

Automotive industries (e.g., PSA[5], Ford[6]) seem to be the leaders of applying the VR technology for design reviews (see Fig. 1.6), which significantly reduce the costs to produce physical prototypes, saving both money and time in the development process, and thus contribute to the development effort to deliver quality products at the right times.

1.3.2.3 VR for Ergonomics Analysis

The ergonomics represents a comfort issue and defines whether a product or an item of work will be convenient to carry out or not. In [Picon, 2010], the author separated the study of the ergonomics into two aspects: from the users' point of view to evaluate whether the product is easy to utilize, and from the workers' point of view to validate whether it is difficult to assembly two pieces. The goal of ergonomics analysis remains the verification of the concept or feasibility before the production of a physical model. In this case, the ergonomics explores whether the contemplated actions for assemblies or the usage of a product are all achieved. For example, from the cognitive point of

[5] www.psa-peugeot-citroen.com
[6] www.ford.com

view, the focus on the human engineer and the acknowledgement on the importance of human input to the design process were explored by [Holt et al., 2004; Brough et al., 2007]. Some other ergonomics evaluations of well defined assembly tasks [Wall and Harwin, 2000; Bordegoni et al., 2009] demonstrated that haptics demonstrate the potential to significantly increase both speed and accuracy of human-computer interactions. The authors of [Pontonnier and Dumont, 2009, 2010] introduced a biomechanical model for the design of motion analysis tools of the upper extremity to improve the ergonomics of workstations in a VR environment.

The above three aspects of VR applications aim at identifying the problematic issues of a product design, and these problems include the infeasibility of assemblying two parts, the defect of the general appearance, and the flaw of ergonomics. Once a design problem is recognized, designers or CAD engineers receive a feedback to modify the design. And then a new evaluation process would be carried out. This procedure of identification, modification and re-design is iterative until a specification is satisfied (see Fig. 1.2). Such physical displacement of project actors and data between the simulation facilities and the design office increases the design time and breaks the continuity of data work-flow. The following aspect of the VR application concerns the combination of a design and an evaluation process in an immersive virtual environment.

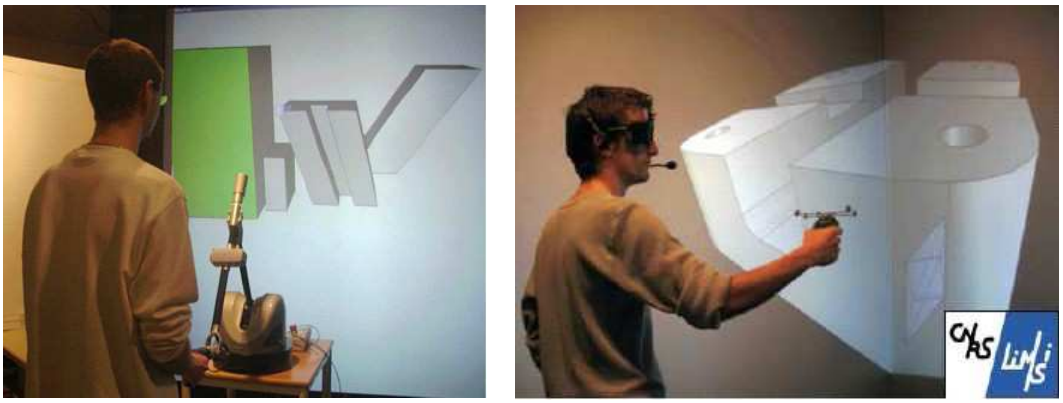
1.3.2.4 VR for Design and Simulation Analysis

Nowadays, CAD systems include functions of simulation allowing, for example, to assess the resistance of materials to the deformation or temperature. However, the interfaces of these CAD systems are less intuitive, efficient and user-friendly comparing with the applications in a virtual environment especially the employment of haptic devices. In this section, we identify some typical applications that employ the VR environment for part design and simulation analysis.

An application termed VRAD was developed in the research group VENISE[7] in the framework of Convard's thesis [Convard, 2005]. The objective of their demonstrator is to propose an immersive design solution through voice commands and body gestures (see Fig. 1.7 (b)). Picon's thesis [Picon, 2010] improved the work of VRAD in terms of providing haptic solutions for the selection and the modification of a digital mock-ups (see Fig. 1.7 (a)).

In [Fischer et al., 2009], the authors combined the procedures of shape deformation and shape re-design together in a $C6$ virtual environment[8] to encourage the rapid investigation of many possible shape design choices (see Fig. 1.8 (a)). A deficiency is that their deformation simulation was derived from a mesh-free method, and thus the deformation accuracy could not be strictly assured. Similarly, authors of [Meyrueis et al., 2009] introduced a 3D object deformation method in an immersive environment dedicated to the development and optimization of industrial parts designed with CAD. This method was divided in three steps (D3): selection by Drawing, Deformation, and a final Design step. The deformations were driven by user's gestures, who sketched the deformation shape (see Fig. 1.8 (b)). These two example applications keep the continuity of data flow as they combine the design and evaluation processes and thus the design decision taken in the immersive environment can be directly reflected in

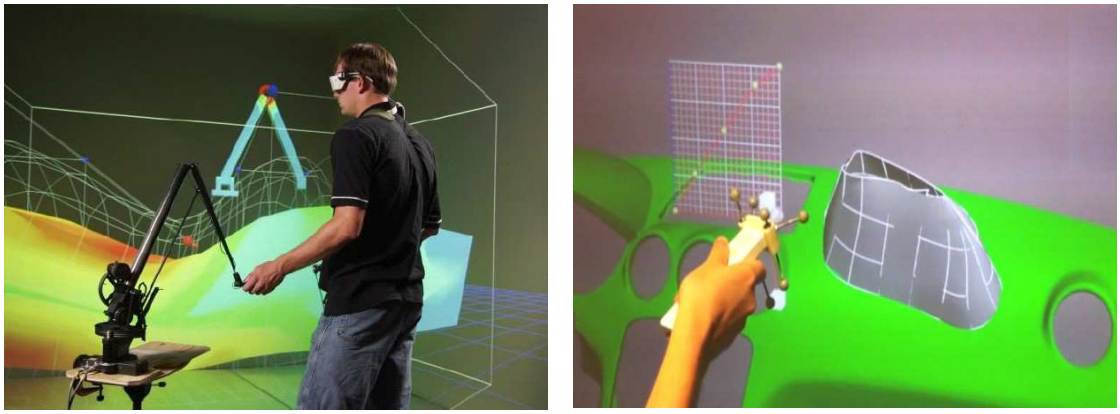
[7] www.limsi.fr/venise/
[8] www.vrac.iastate.edu/c6.php



(a) Extrusion with force feedback (b) VRAD design environment

Figure 1.7 – (a) A user is trying to accomplish a task of extrusion and the scene contains two existing objects in addition to the object being manipulated; and (b) a user is editing of a digital model by voice commands and gestures.

a new design investigation. And moreover, the designer can benefit from the force feedback due to the employment of haptic interfaces which aid to deform a digital model.



(a) Shape deformation (b) Immnersive design

Figure 1.8 – (a) Haptic feedback to guide interactive product design; and (b) an immersive design environment named D3: Draw, Deform and Design.

1.3.3 Preliminary Conclusion

The VR technology has been applied in many industrial applications which aim at saving costs during the implementation of a PLM. A host of research has illustrated that the employment of haptic interfaces in VR environments not only enhances users' performance of completing virtual tasks, but also provides the potential of enriching interaction channels during different stages of part design [Aziz and Mousavi, 2009; Bordegoni and Cugini, 2006]. And therefore the employment of VR and haptic technology contributes to accelerate the decision making process in the overall product development cycle [Yang et al., 2001; Eddy and Lewis, 2002]. However, a large majority of these industrial evaluation applications are focusing on haptics-based assemblies or products maintenance operations. Applications concerning interactive finite element

analysis in a VR environment are not largely explored. In this section, we would like to stress the necessity to improve these existing industrial applications.

- Most of the industrial VR applications lack of a method for the exchange of data among different tools [Zorriassatine et al., 2003]. The integration among design, analysis and simulation tool should be improved. A CAD-VR integration paradigm was proposed in [Bourdot et al., 2010] to implicitly edit mechanical parts through multi-modal immersive interactions. An approach denoted as "Simulated Reality" was introduced by [Stork et al., 2008] aiming at the interplay among processes of simulation, optimization and interactive visualization. However, the main challenge is related with the complexity of CAD models that were not intended to be viewed in real time [Raposo et al., 2009].
- Deformable object modelling is not generally included in VR-PLM integration applications, although 2D deformation cases [Loock et al., 2001] or deformation by spatially distributed control points [Fischer et al., 2009] were considered in few applications . However, these applications either lacked real-time haptic interactions, or did not take the simulation of *volumetric deformable objects* into account.

In this thesis, the goal application is to develop a framework for design evaluations of industrial deformable parts through real-time haptic interactions. On one hand, we simulate volumetric deformations, which are suitable for analysing internal strain or stress. On the other hand, our simulation is coupled with haptic devices and the sensation of haptic feedback contributes to identify of problematic issues of industrial deformable parts by checking deformations. Moreover, the real-time deformation modelling method proposed in the thesis can contribute to assembly applications in terms of integrating deformable objects with rigid assembly scenes. And much further forward, our interactive deformation evaluation framework can also be regarded as a progress towards the integration of CAD, FEA and VR.

1.4 Overview of the Thesis

Although modern CAD systems tend to integrate CAE procedures (e.g., finite element analysis) within an integrated design environment, the generation of a finite element model from a design model remains a tedious task. And unfortunately, CAE simulations lack an intuitive interaction method respecting the user's role during simulations. The research presented in this thesis is motivated by the potential benefits of employing the VR technology as a tool for the PLM. More particularly, the goal of this thesis is to apply haptic interfaces in a VR environment to enhance the design decision-making process by aiding engineers understand better how a deformable mechanical part deforms in the early design phase. And thus the time and costs needed for the evaluation of such mechanical parts can be largely reduced and the design information work-flow in the PLM can be continuous as engineers in the different departments are allowed to manipulate the deformable parts (see Fig. 1.4).

In the following section, we present the motivations of thesis from two aspects. On one hand, we introduce an interactive deformation modelling method by indicating the challenge of volumetric deformation simulations. On the other hand, the deformation

simulation method proposed in this thesis is dedicated to an industrial evaluation case, which is detailed in section 1.4.2.

1.4.1 Challenges of 3D Volumetric Deformations

As we discussed in the previous sections, 2D deformation cases were considered in few industrial applications. The extension from 2D deformation cases to volumetric deformations raises challenges, especially in cases where haptic interfaces are introduced into real-time deformation simulations. The reasons can be explained from two aspects. On one hand, the refresh rate required for haptic rendering loop is at least equal to 1000 Hertz so that a good real-time interaction can be obtained without discontinuities [Burdea, 1996]. On the other hand, physically-based deformable models are preferred to generate high fidelity deformation results. The relations from the theory of elasticity are usually employed to establish the mathematical formulation of the problem, which is finally solved by the finite element method (FEM). Performing FEM computations within a haptic rendering loop is far beyond the capacity of nowadays computers. We argue that the challenge can be effectively resolved by the two-stage method proposed in this thesis. In the following subsections, we explain in details an industrial evaluation case focusing on deformation validation processes through real-time haptic interactions.

1.4.2 Deformation Evaluation: an Industrial Case

This thesis is lunched in the framework of an industrial project. And one of the aim of the project is the real-time design evaluations of deformable mechanical parts for the purpose of identifying structure insufficiencies in the early design phase.

Project Overview:

Our research is a part of EMOA[9] project hosted by the PSA Peugeot Citroën. More particularly, the background of this thesis concerns the deformation evaluation of deformable mechanical parts. The validation process of geometrical and mechanical properties of prototypes is a very important step within the industrial design process. This involves, for example, the verification of the conformity of specifications represented by more and more complex CAD models. Regarding the stamping process in the automotive industries, the verification of the conformity of a polystyrene, from which the mould of the stamping tool will be achieved, is of first importance.

According to industrial practices in the automotive industry, there are two main factors which influence the design verification cost:

- the increasing globalization of manufacturing leads to the fact that polystyrene molds are machined abroad;
- the evaluation involves manipulations by an expert sent abroad to verify, *on-the-spot*, the functionalities of the physical prototypes.

[9]EMOA: Excellence dans la Maîtrise de l'Ouvrant Automobile.

Assumption and Propositions:

In order to reduce the verification costs and to increase the efficiency of data flow in the PLM, we believe that the employment of the VR is a good choice based on the research review from the industrial point of view in the previous sections. In a first step, the expert who carries out the evaluation tasks should handle interactively the CAD model of the designed form. One of these verifications consists in evaluating the deformation of the form structure under varied solicitations. Broadly speaking, the industrial demand is very strong in the aspect of the design evaluation processes, for example, in assembly simulations or in virtual prototyping.

An assumption is made as follow: the CAD model is the basis for such deformation analysis. Particularly, the polystyrene form is well represented by the CAD model. Based on this assumption, the research will aim at allowing the expert to interact with the digital mock-up. To do this, a real scale visualization will be mandatory in order to improve the immersion sensitivity.

Propositions are proposed based on the assumption and the objective of the industrial project. Particularly, we propose a framework that allows real-time haptic interactions with an industrial deformable part. Our framework explores the following advantages of applying the VR technology towards an industrial PLM.

- using virtual prototypes (e.g., CAD models of polystyrene molds in this thesis) to replace the corresponding costly physical models;
- using a VR environment to allow engineers interact with the virtual prototypes;
- using haptic interfaces (e.g., force feedback) to provide engineers possibilities to better understand how a deformable mock-up deforms.

In order to achieve interactive analysis of a deformable model based on the above propositions, it is mandatory to propose a deformation model, for example based on the FEM which is rigorous from the mechanical point of view. However, the deformation computation will have to meet two antagonist requirements. On one hand, a VR application is user-centred which implies that the size of the models that are handled in the VR session must be limited so that the real-time constraint is well respected. And therefore, the digital model must be carefully prepared before it can be handled in a VR environment. On the other hand, the Virtual Prototyping applications are model-centred in order to make the digital model as realistic as possible. The digital mock-up can be represented by different forms: a CAD model, a meshed model and a real-time simulation model.

As the deformation verification processes are different in the reality comparing with the operations in a virtual environment, we pre-defined global and local haptic verification scenarios in a VR environment with the cooperation of PSA Peugeot Citroën. In the following paragraphs, we explain the local verification scenario and the global verification scenario in detail.

Local Deformation Verification Scenario:

In this thesis, the local deformation verification scenario refers to the process of design evaluation of local structures of a deformable mechanical parts. Here, the deformable mechanical part is represented by the CAD model of a polystyrene stamping mould, while the local structures include ribs or stiffeners. The stiffness validation of these local structures are crucial during the industrial stamping process as they strengthen the bearing capacities of the stamping tool in certain working conditions.

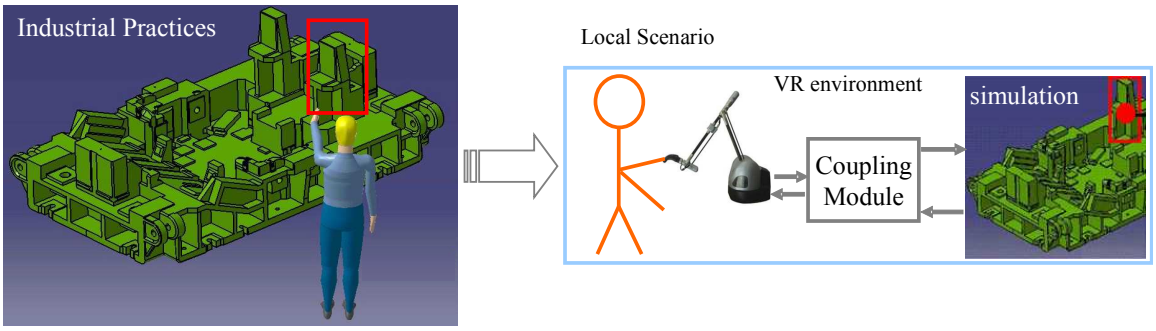


Figure 1.9 – Local deformation verification scenario.

Figure 1.9 illustrates the local deformation verification scenario. The left picture shows the illustration of the industrial practice of validating such local structures by a single hand of an operator. The right picture presents the corresponding paradigm of implementing the local deformation verification scenario in a VR environment with the employment of a haptic interface, which allows an operator to naturally and intuitively interact with the virtual representation of the stamping mould.

Global Deformation Verification Scenario:

After the explanation of the local deformation verification scenario, we focus on the introduction of the global deformation scenario, which refers to the process of design evaluation of the whole deformable mechanical part. The global deformation verification of the stamping tool is equally important compared with the local deformation evaluations, as the surface accuracy of an industrial product, e.g., a car door, can not be guaranteed if the geometrical and mechanical properties of the stamping tool are not fully respected.

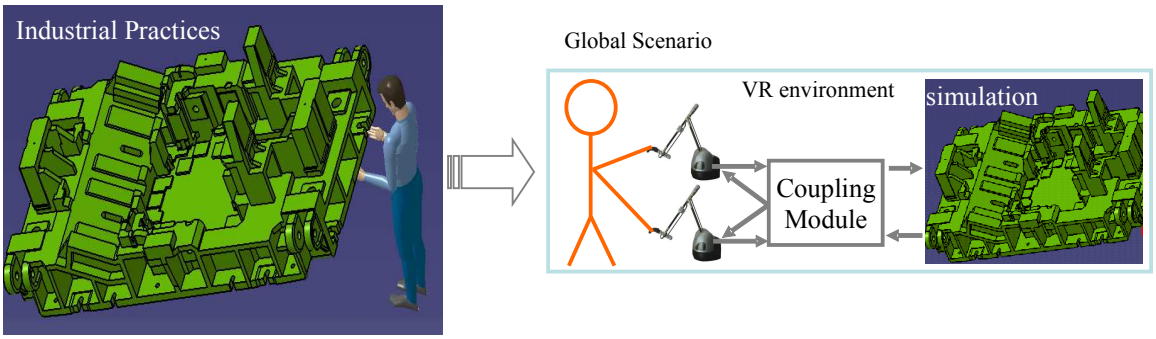


Figure 1.10 – Global deformation verification scenario.

Figure 1.10 demonstrates the global deformation verification scenario. The left picture shows the illustration of the industrial practice of validating the whole structure

by two hands of an operator. Such bimanual evaluation practices are common in automotive industries, and an example is displayed in Fig. 1.3. The right picture shows the corresponding paradigms of carrying out the global deformation verification scenario in a VR environment by introducing two haptic interfaces. During real-time interactions, the movement of one hand is dominant over the other, which is a natural behaviour of human daily-life activities when we accomplish a task required the cooperation of both hands. And moreover, the deformable behaviour of the mechanical parts can be perceived by both hands as the deformation is calculated in a global level. A direct extension of such a paradigm is that two users or more than two users manipulate the same deformable part and they feel each other's influence while touching the part. However, this complicates the overall stability of the simulation system [Barbagli et al., 2003], as multiple users are introduced into the real-time interaction system.

1.4.3 Problem Statement and Thesis Scope

The goal of this thesis is to develop a framework aiming at providing physically-based haptic interaction with deformable industrial mechanical parts. To achieve an accurate deformation result concerning the evaluation scenarios, deformable models formulated by the FEM gain popularities to represent volumetric deformations. And the degree of deformation realism requires a high meshing resolution, which has a direct impact on the size of the matrices involved in the elasticity system and thus results in a heavy computational load in real-time. In order to speed up the interactivity rate while obtaining a trade-off between the deformation accuracy and the real-time haptic interaction performance, there have been several attempts to address the issue of coupling a heavy computation with high refresh rate of a haptic loop. Basically, two categories of solutions are classified: either simplified models based on linearization or reduction are employed, or purposive pre-computations are implemented before the real-time interaction occurs. In the subsequent chapter, we will present a state-of-the-art on haptic interaction with physically-based deformable objects, and we will explain that the model reduction method by a series of pre-computations are suitable to handle the trade-off issue concerning the real-time deformations of complex deformable mechanical parts.

Regarding the scope of this thesis, we are intent to indicate the following two points.

- The addition of deformable objects into haptically assembly procedures is not the interest of this thesis. However, an assembly procedure could benefit from our contributions on real-time volumetric deformation modelling towards haptic interactions.

- The evaluation application of this thesis aims at applying haptic interfaces to guide deformation validations in an interactive way. However, we do not combine the procedures of re-design and re-analysis. An exception reference is [Fischer et al., 2009], in which the authors integrated shape design and deformation analysis together to encourage a rapid investigation of possible design decisions. However, their deformation simulation was derived from a mesh-free method, and thus the deformation accuracy could not be assured strictly.

1.4.4 Thesis Contributions

We demonstrate that the challenge of coupling a heavy computation with high refresh rate of a haptic rendering loop is effectively solved in the evaluation application investigated in this thesis: design validation of deformable mechanical parts. The work presented in this thesis differs from the other previous deformation simulations by either one or all of the following aspects. (1) Instead of local deformations, we simulate global deformations. By "global deformations", we mean deformations that involve the entire body, such as bending or twisting, but not including poking operations. (2) Rather than deformations only involving surface mesh vertices, we simulate the volumetric deformations, which enable us to analyse internal material properties directly determined by the simulation, such as internal material stress or forces.

The contributions of the thesis are summarized as follows, and the implementation details are presented in the subsequent chapters.

- We develop a deformation evaluation framework for real-time haptic interactions by introduce a two-stage method: an off-line phase to pre-compute deformation spaces and an on-line phase to enable haptic interactions by a costless response model.
- We propose a mesh analysis method for the pre-computations during the off-line phase. This method allows the off-line phase to calculate different deformation spaces with an accuracy enhancement regarding correspondent anticipated scenarios. Moreover the method permits a real-time switch among the different pre-computed deformation spaces, so that the on-line deformation computations focus on DOF[10] where necessary.
- We introduce a division scheme to assure the real-time haptic interaction performance. In the scheme, the deformation computation process is divided into two separate modules which are then implemented on two separate threads. One module is dedicated to the update tasks of haptic renderings, which is implemented by extracting a sub-matrix from the pre-computed data matrix, and in this way haptic rendering process can be quickly refreshed at more than 1000 Hertz. The other module is dedicated to the task of deformation computation and visualization.

1.5 Chapter Summary

In this chapter, we explained the importance of applying the VR technology in a PLM, as the traditional CAE tools manifested several disadvantages in terms of interaction methods. Regarding this aspect, we described a representation of the design evaluation cycle within a VR environment by introducing an operator in the interaction loop (see Fig. 1.4). The existing industrial VR-PLM integration applications (e.g., design review, virtual assemblies/disassemblies, ergonomics assessments, etc.) demonstrated the crucial role of the VR in a PLM. However, the primary drawback of these applications was

[10] DOF: degree(s) of freedom

the lack of the management of deformable mechanical parts in a virtual environment. The research work presented in this thesis is toward an interactive project review of deformable mechanical parts through haptic interfaces in a VR environment. By indicating the challenges of real-time haptic interaction with deformation simulations, we briefly presented the main contributions of the thesis at last.

In the next chapter, we will describe the theoretical background on deformation modelling methods and we focus on the modelling methods derived from the finite element method and the theory of elasticity.

Theoretical Background: Deformation Modelling and Finite Element Formulation

2

Contents

COMPUTERS have become an indispensable tool in terms of modelling and simulations. As the computational power increases, users and applications are demanding ever increasing levels of realism in these simulation domains. Particularly, the ability to model and manipulate deformable objects is an essential aspect to many industrial applications. At the same time, however, several challenges arise when interactive or real-time constraints are considered for deformation simulations. In the industrial background, physically-based deformation modelling methods formulated by the theory of elasticity are preferred compared with non-physically based modelling methods, as the former ones describe deformation behaviours in a more rigorous way. However, the trade-off issue between the deformation accuracy and the real-time interaction performance is a primary challenge for the integration between haptic interfaces and deformation simulations formulated by the physically-based modelling method. To handle

the challenge, the model reduction method based on vast pre-computations is popularly employed according to the literature review in the section 3.2 of the third chapter.

In this chapter, we describe the necessary theoretical background for the deformation modelling, including the theory of elasticity, the finite element method formulation and the model reduction method. At last, a brief survey on the current finite element software and solvers is presented, as the deformation modelling methods based on the pre-computations reply heavily on a FEM solver, either free or commercial.

2.1 Deformation Modelling

In the field of computer graphics and computer animations, a plausible deformation modelling is essential to describe the behaviour of the human body, animals, and soft objects. However, it makes a different story for interactive applications (e.g., computer games), as such applications have very limited computing budgets for 3D physical continuum simulations. Concerning the interactive simulation of deformable objects in the background of industrial design evaluations, a major challenge rises from the fact that an operator may interact with deformable objects and receive immediate sensory feedbacks via human-computer interfaces (e.g., force feedbacks). A sound proof is that the current virtual prototyping and assembly planning applications require interactive simulations of deformable models with complex geometries. In this thesis, we investigate industrial deformable parts with an aim to provide both sufficiently realisms to capture the underlying physics, and sufficiently fast to satisfy the interactive and real-time constraints.

Generally speaking, a deformation can be defined as a change in the shape or size of an object due to the forces applied on the surfaces or on the internal positions of the object. And thus the deformation modelling process mainly focuses on the behaviour description of a deformable object: the relations between the applied forces and the deformation of the object. A straightforward approach to the deformation modelling is to create a real solid model. Nevertheless, this approach has several drawbacks, as the production of a physical model is expensive and time-consuming due to the complexity of the manufacturing process. This is why today's deformation modelling approaches and industrial design evaluations primarily reply on computer simulations, which are the essential component for virtual prototyping applications from the industrial point of view.

In the following sections, we briefly overview a series of deformation modelling methods with an emphasis on the physically-based modelling methods derived from the FEM and the theory of elasticity, as the physically-based method can better describe the deformation behaviour of deformable mechanical parts and thus they are suitable for industrial design validation through haptic interfaces.

2.1.1 Overview of Deformation Modelling Methods

The deformation modelling methods can be classified into two main categories which are denoted as non-physically based methods and physically-based methods. Each of them is briefly described below according to the comprehensive overview presented in [Gibson and Mirtich, 1997; Nealen et al., 2005].

2.1.1.1 Non-physically based Deformation Modelling

The non-physically based methods for modelling of the deformable objects are usually based on pure heuristic geometric techniques or apply a sort of simplified physical principles to achieve a plausible visualization result. They have been popular primarily in the computer graphics. Two well-known examples of non-physically based approaches are presented as follows.

Spline deformation modelling method. In this method, the planar, $3D$ curves and surfaces are represented by a set of control points. The shape of complex objects is then modified by varying the position of the control points. Some example applications of using spline modelling method to describe deformations can be found in [Bookstein, 1989; Bazen and Gerez, 2003].

Free form deformation modelling method. The key idea of this method is to deform the shape of an object by deforming the corresponding space in which it is embedded. Examples are available in [Sederberg and Parry, 1986; Coquillart, 1990].

The non-physically based methods are usually fast and efficient to model interactive or real-time deformation animations. However, they are not suitable for modelling the cases where the physically realistic deformation behaviour is desired. In this thesis, non-physically based modelling methods are not adequate to describe the internal material properties, as we focus on design validations of deformable mechanical parts through haptic interfaces.

2.1.1.2 Physically-based Deformation Modelling

The physically-based methods are based on the mathematical models which are generally formulated by a series of partial differential equations (PDEs), which are derived from the theory of elasticity. Therefore, the physically-based deformation modelling methods are capable of representing behaviours of deformable mechanical parts, although the expensive solution computations can become a barrier to employ such methods to model real-time deformations.

Different physically-based deformation modelling methods are presented with a brief description as follows.

Mass-Spring-Damper method (MSD). In this method, the mass is concentrated in a number of nodes which are connected by linear springs. When a force is applied to a node, it starts to move and pass the force via springs to the neighbouring nodes. Some examples of this method can be found in [Cover et al., 1993; Elhelw et al., 2003].

From the computational point of view, the deformation modelling based on the MSD method is simple and the cost of the calculation is low. The major drawback of the method is their insufficient approximation of true material properties. And moreover, they are insufficient in the case where the geometry of the deformable object is complex. It should be noted that some research work have been carried out to improve the deformation accuracy resulted from the MSD method (see the section 3.2.2.1 of chapter 3).

Finite difference method (FDM). In this method, the continuous derivative which appears in the PDE-based formulations is replaced by a finite difference approximation. And this approximation is computed in a series of grid points which are organized regularly spanning over the domain of the object. Some applications can be found in [Sarti et al., 1999; Kaus et al., 2003].

Deformation simulations based on the FDM is accurate and efficient for the objects with regular geometry, as the distribution of computational points should be arranged regularly to obtain a better deformation result. Nevertheless, the discretization would become extremely dense in case of complex shape of the domain, and thereby lead to a huge computational complexity.

Boundary element method (BEM). In this method, the differential problem is converted to the integral form. And under special boundary conditions, the integration over the volumetric domain could be substituted by the integration over its boundary. A deformation model using the BEM was described in [James and Pai, 1999].

The deformation models based on this method cannot deal with any phenomena related to the volume of the body, as the final solution space is dependent on its boundary. For example, the applied volume forces or heterogeneity of the body cannot be considered.

Finite element method (FEM). This method has been widely employed to model deformable objects, as it is directly based on the theory of the elasticity and it provides a better approximation for the complex geometries compared with the FDM. This method generally consists of discretization of the domain and mathematical reformulation of the boundary-value problem, resulting in large systems of algebraic equations. The details of this method together with example applications are presented in the subsequent chapters of the thesis.

The other less known approaches are represented by *Finite Sphere Models* (FSM) presented in [Basdogan et al., 2004] as well as the *Long Element Method* (LEM) proposed in [Balaniuk and Salisbury, 2002]. These examples represent some modifications of FEM to be suitable for certain types of deformation simulation problems.

2.1.1.3 Preliminary Conclusion

To conclude, the physically-based deformation modelling methods are computationally expensive. However, they fully respect the underlying material properties and thus produce high fidelity deformation results comparing with the non-physically based deformation modelling methods. Moreover, the FEM is superior to the other physically-based methods for modelling complex deformable objects with non-trivial physical properties.

Considering the goal application of this thesis: design validation of deformable mechanical parts to satisfy industrial evaluation requirements, we simulate the volumetric deformations which contribute to analyse internal material properties directly determined by the simulation, such as internal material stress or forces. Therefore, deformable models formulated by the FEM are employed in the thesis, as such models fits best the desired properties of the mechanical parts considered here. However, the

employment of FEM models for real-time haptic interactions raises a challenge of handling the trade-off between the deformation accuracy and the real-time performance. In chapter 3, we will present a state of the art (section 3.2) to introduce various methods which are dedicated to deal with the trade-off.

2.1.2 The Theory of Elasticity

The theory of elasticity is a part of broader study known as *continuum mechanics*, where equations describing the behaviour of the continuous media can be generally divided into four major categories: (a) kinematic, (b) kinetic, (c) thermodynamic and (d) constitutive [Oden, 1972]. The thermodynamic principles connected to the heat conduction of the body are not considered in this thesis.

The continuum models can be given by description of relationship between four components: displacement, strain, stress and force. The *displacement*, representing the movement of particles in the continuum, results in a change in small volume element which is measured by *strain*. Similarly as the displacement, the *forces* are applied to the particles of the continuum and their effect on the small volume element is measured by *stress*. Both the displacement and forces can be regarded as external factors, whereas the stress and the strain are the internal measures.

2.1.2.1 Strain and Stress

In this section, we will briefly introduce the fundamental concepts for the description and measurement of the state of deformations.

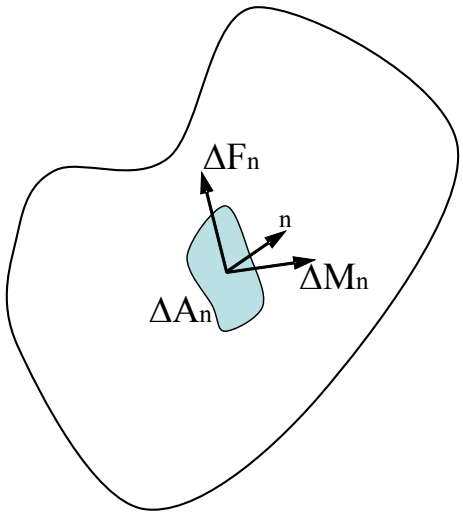


Figure 2.1 – Resultant force and moment on a small oriented area.

Given a deformable object (see Fig. 2.1), consider a small area ΔA_n on the surface of or within the body, which has as orientation specified by unit vector $\vec{\mathbf{n}}$. Let $\Delta\mathbf{F}_n$ and $\Delta\mathbf{M}_n$ be the resultant force vector and moment vector, respectively, exerted on oriented area $\Delta\mathbf{A}_n$. We seek the intensity of the resultants on the oriented area element $\Delta\mathbf{A}_n$ by taking the limit as follows:

$$\lim_{\Delta A_n \to 0} \frac{\Delta\mathbf{F}_n}{\Delta\mathbf{A}_n} = \frac{d\mathbf{F}_n}{d\mathbf{A}_n} = \mathbf{T}_n, \tag{2.1}$$

$$\lim_{\Delta A_n \to 0} \frac{\Delta \mathbf{M}_n}{\Delta \mathbf{A}_n} = \frac{d\mathbf{M}_n}{d\mathbf{A}_n} = \mathbf{C}_n. \tag{2.2}$$

Here, $\mathbf{T}_n$ is the stress vector or traction, $\mathbf{C}_n$ is the couple-stress vector. The elementary theory of elasticity proceeds on the assumption that $\mathbf{C}_n = 0$, while the traction $\mathbf{T}_n$ represents the stress intensity at the point for the particular area element of orientation specified by unit normal vector $\vec{\mathbf{n}}$. A complete description of the stress at the point requires the traction for all directions. Fortunately this problem can be solved by traction in principle directions.

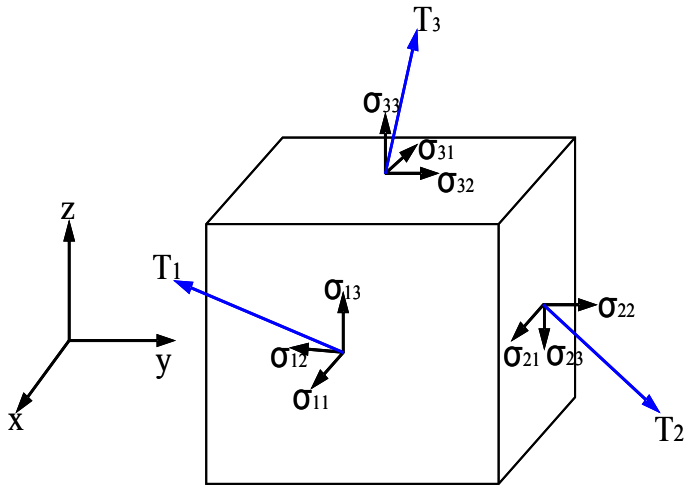


Figure 2.2 – Stress components.

Given a point $X(x, y, z)$ in the deformable body, consider an infinitesimal cube centred at this point. Figure 2.2 shows the three tractions $\mathbf{T}_i (i = 1, 2, 3)$, on the faces with surface normal vectors along the principle directions. Each such traction $\mathbf{T}_i$ has components along the principle axis. The nine stress components (each a scale value) shown in Fig. 2.2 form a second order tensor σ as following.

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{2.3}$$

Stress tensor σ gives a complete description of the internal force intensity at a given point. Similarly, a strain tensor at a point gives a complete description of the state of deformation at that point. The strain tensor ϵ is also second order tensor as following,

$$\epsilon = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix}. \tag{2.4}$$

Due to the symmetry, there are only six independent variables in the stress tensor and the strain tensor. In the rest of this thesis, we will simply use the vector representation for stress and strain, instead of their tensor representation. The vector

representation of stress is,

$$\sigma = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} \tag{2.5}$$

here, the first three components are called *normal stresses*, while the other three are called *shear stresses*.

The vector representation of strain is:

$$\epsilon = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix}. \tag{2.6}$$

Given the point-wise displacement vector $\mathbf{u} = \mathbf{u}(X) = (u, v, w)^T$ at point X of the deformable body, the strain is given as following:

$$\epsilon_x = \frac{\partial u}{\partial x} + \frac{1}{2}\left[(\tfrac{\partial u}{\partial x})^2 + (\tfrac{\partial v}{\partial x})^2 + (\tfrac{\partial w}{\partial x})^2\right], \tag{2.7}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \left[\tfrac{\partial u}{\partial x}\tfrac{\partial u}{\partial y} + \tfrac{\partial v}{\partial x}\tfrac{\partial v}{\partial y} + \tfrac{\partial w}{\partial x}\tfrac{\partial w}{\partial y}\right]. \tag{2.8}$$

And the other four terms of the strain vector is defined similarly. This strain tensor is sometimes also called the *Green-Lagrange strain tensor*, to distinguish it from the linearized version of strain, *Cauchy strain tensor*, which have the following forms. A more mathematically complete development of traction, stress and strain can be found in [Oden, 1972].

$$\epsilon_x = \frac{\partial u}{\partial x} \tag{2.9}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{2.10}$$

An illustration of the difference between the deformable objects employing non-linear and linear strain tensor is shown in Fig. 2.3. In the case of linear strain tensor, the large deformation are obviously not realistic due to the volume growth.

2.1.2.2 Constitutive Equation and Linear Materials

The strain tensors formulated above are defined uniquely based on the local deformation field $\mathbf{u}$ in the vicinity of X. They are purely geometric measures, that is, they do not depend on material properties of the mesh such as moduli of elasticity, compressibility, or similar. So, in order to go from geometric deformations to actual forces, we need to choose a particular *material deformation law* (also called the "constitutive equation").

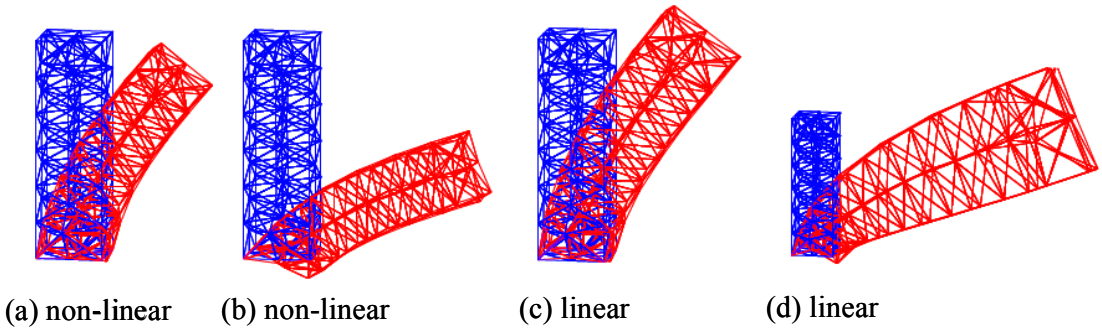


Figure 2.3 – Illustration of the difference between the deformable objects using non-linear strain tensor $((a),(b))$ and linear strain tensor $((c),(d))$. In the case of the linear tensor, the large deformations are obviously not realistic due to the volume growth.

This law will encode physical properties such as material elasticity, compressibility, plasticity, etc.

In this thesis, we model the deformable objects as damped linear elastic material. Namely there is a linear relationship between the stress vector and the strain vector as following,

$$\sigma = \Lambda(\epsilon - \epsilon_0) + \sigma_0 \tag{2.11}$$

where Λ is an elasticity matrix containing the appropriate material properties, and σ and σ_0 are the initial strain and stress, respectively. Usually we assume both initial strain and stress are zero. Therefore we have the following *linear strain-stress* relationship,

$$\sigma = \Lambda\epsilon. \tag{2.12}$$

If the material is isotropic, the elasticity matrix Λ only has two degrees of freedom: Young's modulus E and Poison's ratio ν. And the elasticity matrix λ for isotropic deformable material has the following form,

$$\Lambda = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{2.13}$$

where μ and λ are denoted as *lamé's constants*, and

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}. \tag{2.14}$$

All our simulation experiments throughout the thesis are based on the polystyrene material: Young's modulus $E = 2200N/m^2$, Poisson's ratio $\mu = 0.38$, and material

density $\rho = 1200kg/m^3$.

2.1.3 Preliminary Conclusion

Based on the industrial project discussion in section 1.4.2 and the local and global deformation verification scenarios, we focus on small global deformations and large rigid-body motions in this thesis. By "large rigid-body motion", we mean that the deformable object can be moveable freely and its spatial rigid motion of a deformable object is large compared with the deformation within the object. By "global deformation", we mean deformations that are involve in the entire object, such as bending or twisting. However, the poking operations in small surface regions, which usually happen in surgical simulations [Cotin et al., 1999], is not considered in this thesis. More particularly, the small deformation assumption leads to the linear elasticity model, which is formulated by the linear strain approximation (Equation 2.9 and 2.10). Although the linear strain approximation does not model finite rotations correctly, it contributes to real-time haptic applications in the aspect of the followings. First, the linear strain assumption admits a very high degree of pre-computations which provide a costless real-time deformation response model suitable for haptic interactions. And this is particularly important for applying the VR technology for the virtual prototyping in the early design phase, as during this phase, complex deformation behaviours are not necessarily required. Second, regarding many types of non-linear elastic models, the degree of useful pre-computations which are really need for real-time deformation simulations is quite limited and it is difficult to predict real-time interaction situations. However, linear elastic models are exceptions due to the time-independent pre-computations and the applicability of linear superposition principles for real-time deformation computations. The two-stage method proposed in this thesis takes advantages of the linear assumption of deformable models to satisfy the industrial design evaluation requirements.

2.2 Finite Element Method

In the numerical analysis, the FEM belongs to a class of so called *Galerkin methods* which convert continuous problems to discrete problems. Originally, it was designed for solving the problems in structural mechanics mainly in construction of buildings. Then, it became an excellent tool in many areas of modellings and simulations. As the deformation simulation method proposed in this thesis shares some FEM formulations with a discrete vibration system, the reader can refer to [Shabana, 1990] for a detailed deducing process of FEM formulations. In this section, we will briefly present the discretization process of continuous solids, including the mesh generation, and the equation of motion.

2.2.1 Domain Discretization and Mesh Generation

In this process, the continuous domain Ω is discretized by a triangulation Γ_Ω The triangulation is often referenced as finite element mesh. Each element is given by a set of nodes, edges and faces. The nodes are usually represented by the vertices of

the element. However, in the case of higher-order elements, supplemental nodes are added, e.g., being placed in the middle of the element edges or faces. In the case of two-dimensional domain, the most popular elements are represented by triangles and rectangles (see the $(a), (b), (c)$ and the (d) in Fig. 2.4), whereas tetrahedral, pyramidal, parallelepiped or prismatic elements are commonly used to discretize three-dimensional domains (see the (e) and (f) in Fig. 2.4). A good triangulation of a complex domain is a non-trivial process, as the elements should fulfil some conditions concerning the shape. For example, taking tetrahedral elements into account, *aspect ratio* is computed as a measure of quality, which is usually defined as the maximum side length to the minimum altitude.

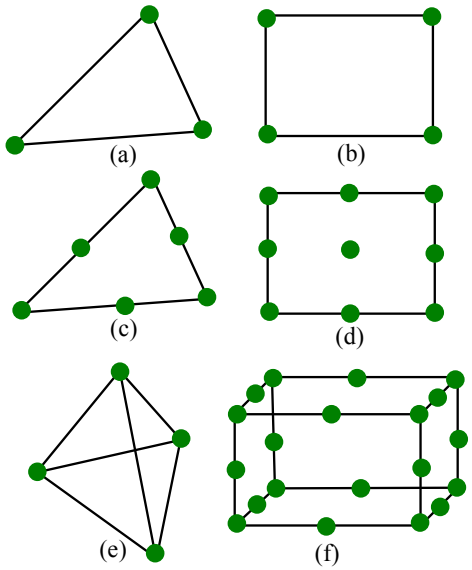


Figure 2.4 – Several $2D$ and $3D$ elements used in FEM: (a) linear triangular element with 3 nodes, (b) linear rectangular element with 4 nodes, (c) quadratic triangular element 6 nodes, (d) Lagrangian element with 9 nodes, e tetrahedral element with 4 nodes, and (f) brick element with 20 nodes.

Of course, the FEM provides several different types of elements. Throughout the thesis, we apply the first-order tetrahedron element (see the (e) in Fig. 2.4), which is the most common element type being used, because this is the simplest kind of a FEM element, with a constant value of strain inside each tetrahedron. The deformation modelling method proposed in this thesis will not change when other types of elements are considered. A particular body deformation is specified by the displacements of mesh vertices, while a small set of vertices are constrained to have zero displacements. For example, for a volumetric mesh model consisting of n vertices, the *displacement vector* $\mathbf{u} \in \mathbb{R}^{3n}$ contains the x, y, z world-coordinate displacements of model vertices.

As our goal application is to develop a framework for design validation of deformable mechanical parts, a mesh model with a sufficient meshing resolution is necessary to achieve an accurate deformation result. However, for haptic interactions with these deformation evaluations, the trade-off between the deformation accuracy, which is generally determined by the density of triangulation, and the real-time interaction performance has to be obtained. Regarding this point, we propose, in this thesis, an off-line mesh analysis method based on anticipated deformation validation scenarios. The method pre-computes different modal deformation spaces off-line and enables

the real-time deformation computation to be switched with respect to operators' interactions. Our mesh analysis method can be viewed as an attempt to enhance the deformation accuracy on DOFs where are most important. Moreover, this method can be implemented regardless of the particular choice of the element type, although all our simulations through the thesis are based on the first order tetrahedron element.

2.2.2 The Equation of Motion

The unreduced equations of motion for structural vibrations of a volumetric $3D$ deformable object, under the FEM discretization, can be derived from the principle of virtual work of Lagrangian mechanics. These equations of motion, normally denoted as *the Euler-Lagrange equation*, are a second-order system of ordinary differential equations [Shabana, 1990].

$$M\ddot{\mathbf{u}} + D(\mathbf{u}, \dot{\mathbf{u}}) + R(\mathbf{u}) = \mathbf{F} \tag{2.15}$$

Here, $\mathbf{u}$ $(u, v, w) \in I\!R^{3n}$ is the displacement vector (the unknown), $M \in I\!R^{3n,3n}$ is the mass matrix, $D(\mathbf{u}, \dot{\mathbf{u}})$ are damping forces, and $R(\mathbf{u})$ are internal deformation forces, and $\mathbf{F} \in I\!R^{3n}$ is the external forces resulting from user interactions or collision reaction forces. The mass matrix is constant in time and depends only on the object's mesh and mass density distribution in the undeformed configuration. In general, it is a sparse non-diagonal matrix, however for algorithmic convenience, it is often simplified into a diagonal matrix by accumulating all the row entries onto the diagonal element [Zhuang and Canny, 2000]. Such lumping essentially means that all elements re-assign their volumetrically distributed mass to their vertices: it is as if the model consisted of a point-like mass at every simulation vertex, with zero mass anywhere else inside the elements. Such a construction of course means losing some simulation accuracy. It is true that the accuracy loss is smaller with finer meshes.

For haptic applications, an explicit employment of the FEM leads to huge computations and thus hinder the real-time interaction performance. One simplified method is based on small deformation assumption by employing the linear elasticity theory (see Equation 2.9 and 2.10). This linear strain makes the internal force vector linear with respect to nodal displacement vector. Namely, it simplifies Equation 2.15 to the following *linear* system:

$$M\ddot{\mathbf{u}} + D\dot{\mathbf{u}} + K\mathbf{u} = \mathbf{F}. \tag{2.16}$$

As the stiffness matrix and the mass matrix are keeping constant during real-time deformation simulations, the above simplified linear system permits an efficient pre-computation process, which is a key to guarantee the real-time interaction performance.

As in the thesis, we focus on the deformation validation of *complex* industrial mechanical parts, the dynamic solving of time-dependent differential equations is not suitable for introducing haptic devices into these applications. In order to speed up the interactivity rate while keeping the accuracy advantages of the FEM models, we propose a two-stage method combining an off-line pre-computation phase and an on-line deformation interaction phase. And our deformation modelling method takes advantage of the linearity of the equations to perform a pre-computation process based on the model reduction method described in section 3.2.3.

2.2.2.1 Special Cases

Sometimes, one is interested only in the deformations assumed under a certain fixed static load, as opposed to the dynamics of deformations. Such static simulations are useful, for example, when determining how structures are able to sustain applied loads. In this section, we would like to point out two special cases of the equation of motion.

One special case is the system of equations for static deformations. It is obtained by setting the nodal accelerations and velocities to zeros. This leads to the following system of equations for static global deformation,

$$R(\mathbf{u}) = \mathbf{F}. \tag{2.17}$$

Another special case is the corresponding linear system for static small deformation is written as,

$$K\mathbf{u} = \mathbf{F}. \tag{2.18}$$

In this thesis, our two-stage method based on the model reduction method can be applied equally well to these special cases concerning static deformation simulations: simply discarding the reduced mass matrix and damping terms.

2.3 Model Reduction and Modal Analysis

The discretization process of a physically-based model generally results in a system with huge DOFs, as the state of a deformable object is characterized by the positions and velocities of a large number of mesh vertices. And therefore the system is slow to simulate, limiting their applications in an interactive or a real-time configurations. Alternative approximate continuum models have been proposed to restrict the deformable object to many fewer DOFs, sacrificing generality or accuracy for speed.

2.3.1 Basic Formulation

As discussed by James and Pai [James and Pai, 2004], given N un-deformed point locations $\mathbf{p} = [p_1, p_2, ..., p_N]^T$, a reduced deformation model approximates deformed point locations $\mathbf{p}'$ by a linear superposition of M displacement fields (the columns $\mathbf{U}_i$ of $\mathbf{U}$ in Equation 2.20). The amplitude of each displacement field is given by the reduced coordinates $\mathbf{q}$, as shown in Fig. 2.5,

$$\mathbf{p}' = \mathbf{p} + \mathbf{u}, \tag{2.19}$$

$$\mathbf{u} = \mathbf{U}\mathbf{q}. \tag{2.20}$$

Here $\mathbf{U} = \{U_1, U_2, ..., U_i, ..., U_r\} \in I\!R^{3n,r}$ is some *displacement basis matrix* or known as *deformation sub-space*, specifying a generalized time-independent deformation basis of some $r-$dimensional ($r << 3n$) space of $I\!R^{3n}$. And $\mathbf{q} = \{q_1, q_2, ..., q_r\}^T \in I\!R^r$ is the *reduced displacement vector*, while $\mathbf{u}$ is the *spatial displacement vector*.

It is true that there is an infinite number of possible choices for this sub-space and for the corresponding basis. Good sub-spaces are low-dimensional spaces in which typical

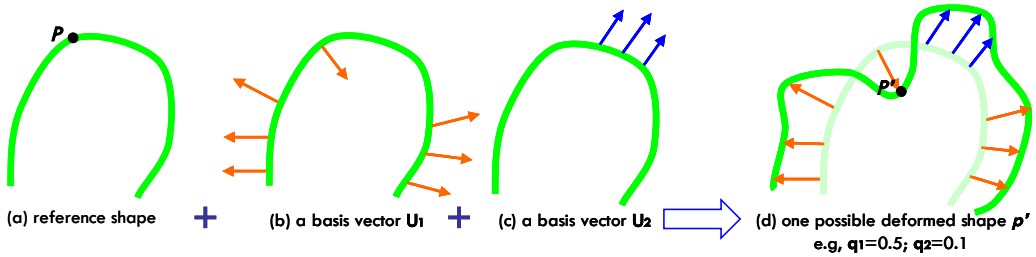


Figure 2.5 – Reduced deformation models: (a) reference shape $\mathbf{p}$, (b) displacement field U_i, (c) displacement field U_j and (d) one possible deformed shape $\mathbf{p}^{'} = \mathbf{p} + U_i q_i + U_j q_j$.

deformations are well approximated. The choice of deformation sub-spaces depend on the geometry, boundary conditions and material properties of a deformable object.

2.3.2 The Reduced Equation of Motion

As in this thesis small global deformations are concerned for the design evaluation of industrial deformable parts, we will focus on the linear version of the equation of motion, which admits a high degree of pre-computations and thus lead to a quick real-time deformation response model suitable for haptic interactions. In [James and Pai, 2004] it is stated that the columns of $\mathbf{U}$ could be obtained from any possible careful designed pre-computations. Selection of a good subspace is a non-trivial problem, for now, we could assume that a good subspace is available, and that it is specified explicitly by some deformation basis matrix $\mathbf{U}$.

By inserting $\mathbf{u} = \mathbf{U}\mathbf{q}$ into Equation 2.16, and pre-multiplying by $\mathbf{U}^T$, we could obtain *the reduced equation of motion*. These equations determine the dynamics of the reduced coordinates $\mathbf{q} = \mathbf{q}(t) \in I\!R^r$, and therefore also the dynamics of $\mathbf{u}(t) = \mathbf{U}\mathbf{q}(t) \in I\!R^{3n}$.

$$\tilde{M}\ddot{\mathbf{q}} + \tilde{D}\dot{\mathbf{q}} + \tilde{K}\mathbf{q} = \tilde{\mathbf{f}} \tag{2.21}$$

where $\tilde{M}\ddot{\mathbf{q}}$, $\tilde{D}\dot{\mathbf{q}}$ and $\tilde{K}\mathbf{q}$ are $r-$dimensional reduced forces,

$$\tilde{\mathbf{f}} = \mathbf{U}^T F. \tag{2.22}$$

As stated in [Barbič and James, 2005], the existence theorem for systems of ordinary differential equations (ODEs) [Hairer and GerhardWanner, 2004] assures that the system of ODEs in Equation 2.21 has a well-defined unique solution, given a specific instance of initial conditions and time-dependent external forces. Since $r << 3n$, the integration of Equation 2.21 is much faster than the integration of the un-reduced system represented by Equation 2.15, albeit with some accuracy loss.

2.3.3 Formulation of Linear Modal Analysis

Pentland and Williams [Pentland and Williams, 1989] developed a simplified expression for the dynamics of deformable objects using modal analysis. Given the mass, damping and stiffness matrices M, D and K, Equation 2.16 can be decoupled into $3n$ linearly

independent ODEs by solving the generalized eigenvalue problem.

$$M\Phi\Lambda = K\Phi \tag{2.23}$$

such that $\Phi^T M\Phi = I$ and $\Phi^T K\Phi = \Lambda$ are diagonal matrices. The entire of Λ contains the eigenvalues, and the columns of $\Phi = \{\varphi_1, \varphi_2, ..., \varphi_i, ..., \varphi_N\}$ contain the eigenvectors of $M^{-1}K$. And Φ is usually termed the *modal matrix* or *modal displacement matrix*, in which i^{th} column φ_i represents the i^{th} mode shape, and the i^{th} entry of Λ is proportional to the resonant frequency of the φ_i. The columns of Φ form a deformation basis of $3n-$dimensional space. And intuitively, the modal matrix forms a natural basis to describe all possible deformations for a linear elastic model.

For volumetric deformation simulations, the six of the vibrational modes account for rigid body motion (object position and orientation). In many applications, it is not necessary to choose all of the vibrational modes to describe the behaviour of a deformable object. as the lowest natural frequencies and their corresponding shapes provide a reasonable description of the deformations of the object (see Fig. 2.6). Higher order modes generally manifest smaller displacement amplitudes and their resonant frequencies can be higher than simulation frame rates. In order to reduce the number of DOFs in the ODEs system to enable a faster simulation, one can disregard higher frequency modes while choosing the number of modes with respect to the simulation requirements. It is also possible to order the vibrational modes so that, in situations where speed is critical, only the rigid body, linear strain, and quadratic strain modes are simulated, and, in situations where accuracy is important, additional motion details can be added systematically by including more higher frequency modes in the analysis. Thus, a modal representation is ideal for controlling level of detail (LOD) in deformable object simulations.

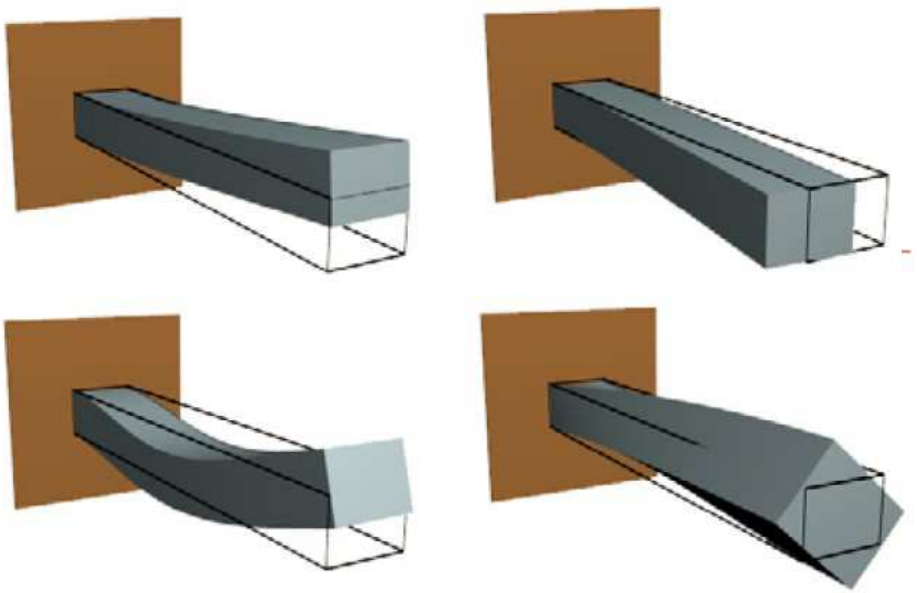


Figure 2.6 – The first four dominant low frequency mode shapes of a constrained beam [Barbič and James, 2005].

Substituting $\mathbf{U}$ by Φ in Equation 2.21, the coefficients of general terms of forces in

a $r-$dimension space can be re-written as follows:

$$\tilde{M}_\Phi = \Phi^T M \Phi = diag(m_i), \tag{2.24}$$

$$\tilde{D}_\Phi = \Phi^T D \Phi = diag(d_i), \tag{2.25}$$

$$\tilde{K}_\Phi = \Phi^T K \Phi = diag(k_i), \tag{2.26}$$

and the modal force $\tilde{\mathbf{f}}_\Phi$.

$$\tilde{\mathbf{f}}_\Phi = \Phi^T \mathbf{F} = \{f_1, f_2, ..., f_i, ..., f_r\}^T \tag{2.27}$$

Here $\tilde{M}_\Phi$, and $\tilde{K}_\Phi$ are diagonal matrices. However, for general damping, $\tilde{D}_\Phi$ is dense. But if we assume proportional (Rayleigh) damping $D = \alpha M + \beta K$, then $\tilde{D}_\Phi = \alpha I + \beta \Lambda$ is also diagonal (see Equation 2.28). Here, α and β are material-dependent positive scalars [Shabana, 1990]. We can now decouple Equation 2.21 and express them by $3n$ independent scalar $2nd$ ODEs as in Equation 2.29.

$$d_i = \alpha m_i + \beta k_i \tag{2.28}$$

$$m_i \ddot{q}_i + d_i \dot{q}_i + k_i q_i = f_i, \quad i = 1, 2, ..., r \tag{2.29}$$

Here the scalar values m_i, d_i and k_i are the modal mass, the modal stiffness and the modal damping of the i^{th} linear mode, respectively. The force $\mathbf{f} = \{f_1, f_2, ..., f_r\}^T$ is the *modal force vector*, which results from a projection of F on a modal sub-space Φ. By defining the un-damped natural frequency of vibration ω_i (in radians),

$$\omega_i = \sqrt{\frac{k_i}{m_i}} \tag{2.30}$$

and therefore the i^{th} diagonal element of Λ termed λ_i, can be written as following.

$$\lambda_i = \frac{k_i}{m_i} = \omega_i^2 \tag{2.31}$$

By inserting Equation 2.28 into Equation 2.29 and substituting the corresponding terms in Equation 2.30 and 2.31, we obtain another form of the reduced equation,

$$\ddot{q}_i + (\alpha + \beta\lambda_i)\dot{q}_i + \lambda_i q_i = \frac{f_i}{m_i} \tag{2.32}$$

after the reduced displacement vector $\mathbf{q}$ is obtained from Equation 2.29, the displacement vector $\mathbf{u}$ can be received by back-substitutions in Equation 2.20.

As detailed in [Hauser et al., 2003], each of these equations in Equation 2.32 has an analytical solution of the following form,

$$q_i = c_1 e^{r_1 t} + c_2 e^{r_2 t} \tag{2.33}$$

where the constants c_1 and c_2 depend on on the initial conditions, and the root of the

characteristic equation $r^2 + (\alpha + \beta\lambda_i)r + \lambda_i = 0$ are

$$r_{1,2} = \frac{-(\alpha + \beta\lambda_i) \pm \sqrt{(\alpha + \beta\lambda_i)^2 - 4\lambda_i}}{2}. \tag{2.34}$$

The solution depends on the sign of $R = (\alpha + \beta\lambda_i)^2 - 4\lambda_i$: $R > 0$, $R = 0$ and $R < 0$ produces the over-damped (r_1, r_2 real and different), critically damped (r_1, r_2 real but repeated) and under-damped case (r_1, r_2 are complex conjugates) respectively. And therefore in any case, q_i in Equation 2.32 will be a real value. The time evolution process of Equation 2.29 and 2.32 will be presented in details in the subsequent chapters.

Similarly, the special case of static linear deformation equation (see Equation 2.18) can also be decoupled by employing modal deformation space Φ,

$$k_i q_i = f_i, \quad i = 1, 2, ..., r \tag{2.35}$$

and thus the motion component q_i can be computed,

$$q_i = \frac{f_i}{k_i}, \quad i = 1, 2, ..., r. \tag{2.36}$$

2.3.4 Preliminary Conclusion

To conclude, we can examine the eigenvalues, discard high frequency mode shapes and thereby only employ dominant modes (i.e., dominant columns φ_i as shown in Fig. 2.6) with the decoupling of original ODEs based on the modal analysis. This process can significantly reduce the real-time computational cost, and thus contributes to the haptic interaction with deformation simulations. In this thesis, the modal parameters introduced in the above equations are pre-computed during an off-line phase. And thus during the real-time interaction phase, the motion components q_i of an individual mode is computed independently and combined by the principle of linear superposition. Our experiments show that linear modal eformation simulations are fast and easily interactive for real-time haptic applications. The choice of choosing deformation mode depends on the simulation feature. For example, low frequency modes are chosen if a global deformations is considered, while high frequency modes are chosen concerning local deformations.

2.4 FEM Software and Solvers

In this section, we present a brief survey on current implementations of methods employed by the FE modelling with reference to [Peterlík, 2009], as our off-line pre-computation phase is based on an open-source mesh generator and a FEM solver.

2.4.1 General Notes on FE Software

Before presenting the concrete implementations of software for FE analysis, some general comments are given.

Nowadays, there are considerable FE softwares ranging from standalone applications to library interfaces. Among them the dominant are commercial packages such as

Ansys[1], Marc[2], Abaqus[3], Nastran[4] and others. Each of this package provide robust and complex algorithms for solving of various problems using the FEM. Nevertheless, they are intended mainly for interactive work, when some particular, usually complex problem is to be solved under special conditions. The other problem concerning the commercial packages is the high purchase cost. Further, the commercial packages are usually presented as black-boxes, so the underlying algorithms cannot be studied. As our goal is to find a software, which can be employed as an engine for repeated computation of relatively simple FE problems. For these reasons, we decided to focus on open-source alternatives presented mainly by libraries providing the interface to the FE routines.

In the FE analysis, the procedures of mesh generation, assembling the FE matrices and solving the FE system are closely interconnected. Nevertheless, unlike the commercial packages, the open-source FE software usually does not include some high-quality mesh generator or high-performance linear solver, but some of the third-party packages are interfaced. This separation becomes slightly problematic, if parallelization of the FEM is desired.

2.4.2 Mesh Generation Software

The first phase which must be performed before the FE analysis takes place is mesh generation. The programs are basically divided into two categories: structured and unstructured mesh generators. Roughly, *a structured mesh* can be recognized by all interior nodes of the mesh having an equal number of adjacent elements. In the area of FEM, unstructured meshes are usually employed. Today's un-structured mesh generators are usually based on some triangulation methods such as *Delaunay triangulation* or *Voronoi diagrams*, where several extension and improvements have been proposed [Shewchuk, 2000].

Well-known mesh generator is represented by an open-source tool TetGen[5], which generates tetrahedral meshed based on the constrained Delaunay tetrahedralizations. The underlying algorithms used for the mesh generation and refinement can be found in [Si and Gärtner, 2005; Si, 2006]. Similar program is represented by NetGen[6], which is also an automatic $3D$ tetrahedral mesh generator. Both TetGen and NetGen tools are interfaced in GMSH[7] program, with built-in CAD engine enabling the user to create interactively the geometry of the object which is to be meshed. It is also equipped with pre-processing and post-processing facilities, so some simpler FE analysis can be performed directly in the GUI of the program [Geuzaine and Remacle, 2008]. For these advantages, in this thesis, the meshing process during an off-line pre-computation phase is based on GMSH.

[1] http://www.ansys.com/
[2] http://www.mscsoftware.com/products/marc.cfm?Q=131&Z=396&Y=400
[3] http://www.simulia.com/products/abaqus_fea.html
[4] http://www.nastran.com/website/files/analysis/analysis2.htm
[5] http://tetgen.berlios.de/
[6] http://www.hpfem.jku.at/netgen/
[7] http://geuz.org/gmsh

2.4.3 FEM Software and Libraries

In this section, we present a brief survey of the open-source packages used for the FE analysis, as commercial FEM packages are not suitable due to their disadvantages discussed above.

A finite element differential equations analysis library *deal.II* written in C++ offers a wide range of procedures for FE modelling [Bangerth et al., 2007]. It supports one, two and three space dimensions, as well as local mesh *h*- and *p-refinement*. It also interfaces other packages such as parallel linear algebra package *PETSc* or the *METIS* tool, enabling the parallelization of the procedures. An advantage of the library is in the extensive and quite intuitive documentation and with good support provided by the authors. Nevertheless, it implements only discretization made up of linear, quadrilateral, and hexahedral cells. However, triangles and tetrahedra are not supported.

FELyX[8] is an object oriented FE code written in C++. It implements the most common elements for solid state mechanics and interfaces some commercial packages. It contains specialized element formulation and new failure criteria for composite materials. Nevertheless, it is focused mainly on the linear models which appear in industrial simulations.

A slightly different concept is implemented by *FEniCS* package[9], a free software for automated solution of differential equations. It provides software tools for working with computational meshes, finite element variational formulations of PDEs, ODE solvers and linear algebra. It is focused on automating the finite element method [Dupont et al., 2003] and it is capable of automatic discretization of differential and integral equations, solution, error checking and optimization. The software can be regarded as a collection of tools and programs, which can be used for various types and stages of FE analysis. It is suitable mainly for experimenting with the finite element methods or design of new models and techniques for finite element analysis.

FreeFEM++[10] is an implementation of a language which is dedicated to the finite element method. It is written in C++ and the *FreeFem++* language is a C++ idiom. It allows for fast prototyping of boundary-value problems. It also supports interpolation of data on several meshes and their manipulation within one program.

The *Getfem++* project[11] focuses on the development of a generic C++ library for finite element methods. Beside the C++ interface, there are also Python and Matlab bindings available together with the library. The goal is to provide a library allowing the computation of any elementary matrix (also for mixed finite element methods) on large class of methods and elements [Renard and Pommier, 2008]. It employs a generic matrix template library *Gmm++* for storing and manipulating the tensor, matrix and vector data. Linear solvers such as MUMPS and SuperLU are interfaced. However, some linear and non-linear solving methods are provided directly by the library. *Getfem++* provides a concepts of bricks template classes which facilitate implementation of various types of problems, materials, boundary conditions, etc. It also supports parallel processing. The main disadvantage of the library is represented by insufficient

[8] http://felyx.sourceforge.net/index.html
[9] http://www.fenics.org
[10] http://www.freefem.org/ff++/index.htm
[11] http://home.gna.org/getfem/

documentation and some non-standard coding conventions.

The *libMesh*[12] library provides a framework for the numerical simulation of partial differential equations using arbitrary unstructured discretizations on series and parallel platforms [Kirk et al., 2006]. It interfaces third-party programs as *BLAS*, *LAPACK* and *PETSc*. It also supports a wide range of elements and shape functions.

The *OOFEM*[13] is free finite element code with object oriented architecture for solving mechanical, transport and fluid mechanics problems that operates on various platforms [Patzak and Bittnar, 2001]. It supports many analysis procedures for structural analysis and advanced modelling features. It also interfaces programs as *PetSC* and *parMETIS*, allowing for parallel processing. Moreover, it implements a full restart support, i.e., the kernel supports full restart from any previously saved state.

Finally, *Code-Aster*[14] is an open source FEM module, which provides, well beyond the standard function of any other thermo-mechanic calculation code, a whole array of analysis methods and multi-physic models. Its application domain extends from seismic analysis, to porous medium, without forgetting acoustics, fatigue, stochastic dynamics. Its models, its algorithms and its solvers have improved in robustness and thoroughness. The main advantage is that it is completely open, and it is easy to use (direct download page for Windows[15]). According to our employment, a meshed part resulting from GMSH is one input for the FEM module in Code-Aster. The other input is a script file containing a sequential of commands of loading mesh, defining material properties (i.e., density, Young's modulus, Poisson's ratio, etc.), specifying boundary conditions, implementing linear modal analysis, and finally outputting results, which are included modal parameters and modal matrix Φ. In the thesis, we choose Code-Aster as a FEM solver for the off-line deformation space computations.

2.5 Chapter Summary

In this chapter, we presented the theoretical formulations of the FEM, the elasticity theory, the model reduction method together with the modal analysis and at last, an overview of popular finite element software and solvers for the purpose of pre-computations is summarized.

To conclude, the physically-based deformation modelling methods (e.g., the FEM) formulated by the theory of elasticity manifest advantages to describe high fidelity deformation results. Concerning haptic guided deformation evaluation applications in the industrial background, the obtention of the trade-off between the deformation accuracy and the real-time performance is crucial when complex mechanical parts are considered. The model reduction method provides an opportunity to spread the width of industrial evaluation applications by coupling deformation simulations based on the FEM with the high refresh rate of the hatic rendering loop.

In the following chapter, a state of the art concerning haptics in virtual reality and physically-based haptic interaction with deformable objects are presented in detail.

[12] http://libmesh.sourceforge.net/
[13] http://www.oofem.org/en/oofem.html
[14] http://www.code-aster.org/
[15] http://www.necs.fr/gb/telechargement.php

Related Work: Haptics in Virtual Reality and Physically-based Haptic Interaction with Deformable Objects

3

Contents

HAPTIC technology refers to the technology that interfaces to a user via the sense of touch by applying forces, vibrations, and/or motions. This mechanical stimulation can be employed to assist in the creation/manipulation of digital mock-ups. Concerning the process of product development in a VR environment, the haptic feedback helps engineers to identify design problems earlier before a physical prototype is manufactured. These problems may range from difficulties in product assemblies to product maintenance procedures. The recognition of these problems during the early design phase can decrease the time and costs required for the product lifecycle management, However, the design evaluation of industrial deformable parts is not popularly explored in a VR environment, as the real-time haptic interaction with volumetric deformations formulated by the finite element method is a challenging task.

In this chapter, we present the foundations of haptics in a VR environemnt followed by a survey of related work which are dedicated to deal with the trade-off issue between the deformation accuracy and the real-time performance.

3.1 Haptics in Virtual Reality

As we described in Chapter 1, the employment of haptic interfaces has been recognized as an essential component for the VR-PLM integration, since these interfaces provide an intuitive channel for the interactions between engineers and simulations in a virtual environment. We present, in this section, the basic foundations of haptics in a VR environment.

Haptic[1] perception covers the general sense of touch and, unlike other senses having localized receptors, is spread over the whole body. Moreover, haptic perception generally involves contacts of the body with objects of the world and allow users to experience a sensation of touch and force feedback when they interact with virtual material in virtual environments [Burdea, 1996]. Traditionally, the term haptic is used to cover two aspects of touch perception: tactile perception involving the perception of sensations at the surface of the skin, and kinesthetic perception involving the perception of body positions and forces [Loomis and Lederman, 1986].

Currently, multiple disciplines such as bio-mechanics, neuroscience, psychophysics, robot design and control, mathematical modeling and simulation, and software engineering converge to support haptics. Wide varieties of applications have emerged and span many areas of human needs, such as product design, medical trainers, and rehabilitation, as we described in the previous sections.

Following the classifications in [Srinivasan and Basdogan, 1997], there are three main areas for haptics research.

- Human haptics – the study of human sensing and manipulation via touch.
- Machine haptics – the design, construction and usage of machines to replace or augment human touch.
- Computer haptics – algorithms and software associated with generating and rendering the touch and feel of virtual objects.

This thesis focuses on the third area of the computer haptics, as it contributes to haptic rendering of realistic deformations and thus guides and strengthens the applications of industrial design evaluations [Fischer et al., 2009]. Nevertheless, since both the human and machine haptics provide information necessary for development of methods and algorithms in computer haptics, they are briefly summarized below. Then, the computer haptics is described focusing mainly on haptic rendering.

3.1.1 Human Haptics

The human haptics is focused on the touch perception in humans. The modality of touch encompasses several distinct sensory systems [Weiner et al., 2003]. Most researcher distinguishes between two systems as presented below using the terminology introduced in [Loomis and Lederman, 1986]. In both cases, the inputs are received

[1]The word "haptic" comes from the Greek "haptesthai" meaning "touch" or "contact". This word was first employed in the 19th century by the historian Riegl and popularized more recently in the field of perception by Gibson [Gibson, 1966].

from mechanoreceptors, specialized nerve endings that respond to mechanical stimulation such as force.

Cutaneous system receives sensory inputs from mechanoreceptors that are embedded in the skin (see Fig. 3.1). The stimuli results in *tactile perception*, producing the sensation of touch. Currently, four mechanoreceptors are known [Longstaff, 2005]: *Ruffini's corpuscles* detecting sustained pressure, *Meissner's corpuscles* detecting changes in texture, Pacinian corpuscule detecting deep pressure and rapid vibrations and Merkel's discs detecting sustained touch.

Kinesthetic system receives sensory inputs from mechanoreceptors collectively denoted as *proprioceptors*, which provide information mainly on angular position of joints as well as tension of muscles and tendons. The stimuli produce sensation that indicates whether the body is moving with required effort, as well as where the various parts of the body are located in relation to each other.

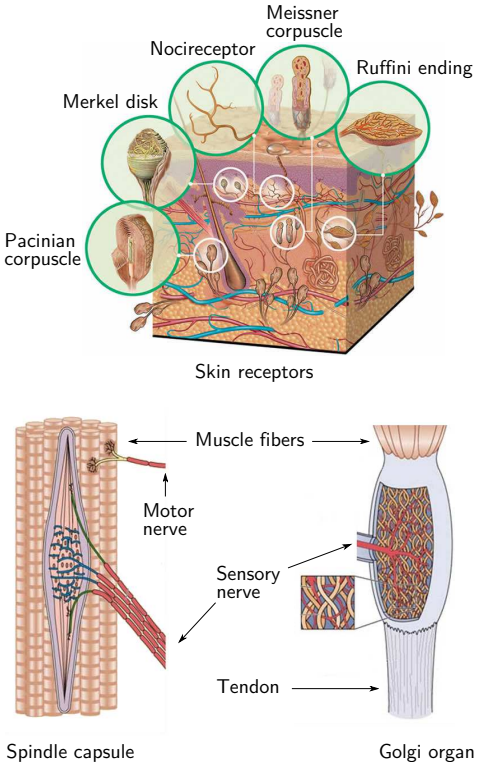
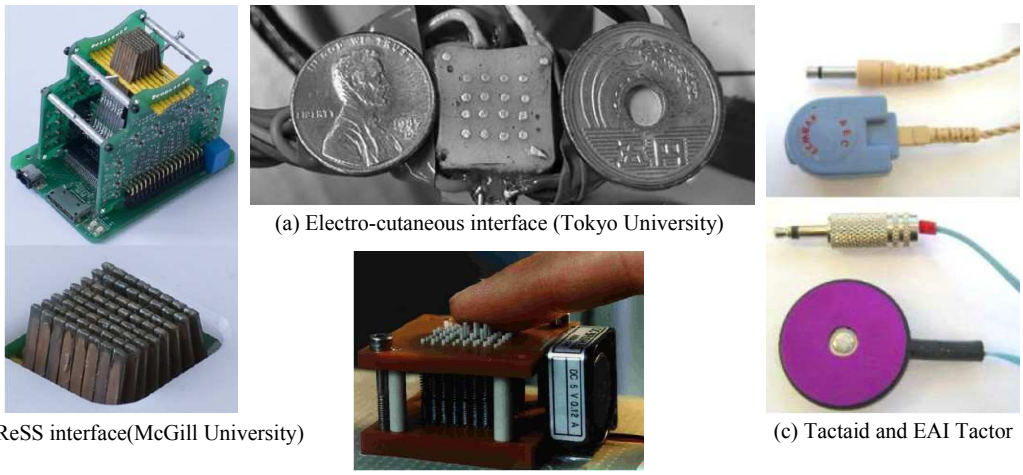


Figure 3.1 – Sensory receptors.

Currently, the human haptics results obtained mainly in the area of psychology and neurology have been successfully applied in both machine and computer haptics. The pioneering work in this area can be found in [Sutherland, 1965] where the vision of a computer driven force feedback device was presented. Another important study is represented by [Lederman and Klatzky, 1997] analyzing the haptic perception as a guide in motion control.

Recently, many phenomena concerning the human touch have been studied, e.g. the ability to distinguish the concave and convex shapes or material properties of objects [Tiest and Kappers, 2007; van der Horst and Kappers, 2008] or haptic illusions described in [Lécuyer et al., 2001; Gosline et al., 2002].



(a) Electro-cutaneous interface (Tokyo University)

(b) STReSS interface(McGill University)

(c) Tactaid and EAI Tactor

(d) Fingerpad (Forschungszentrum Karlsruhe)

Figure 3.2 – Various tactile interfaces.

3.1.2 Machine Haptics

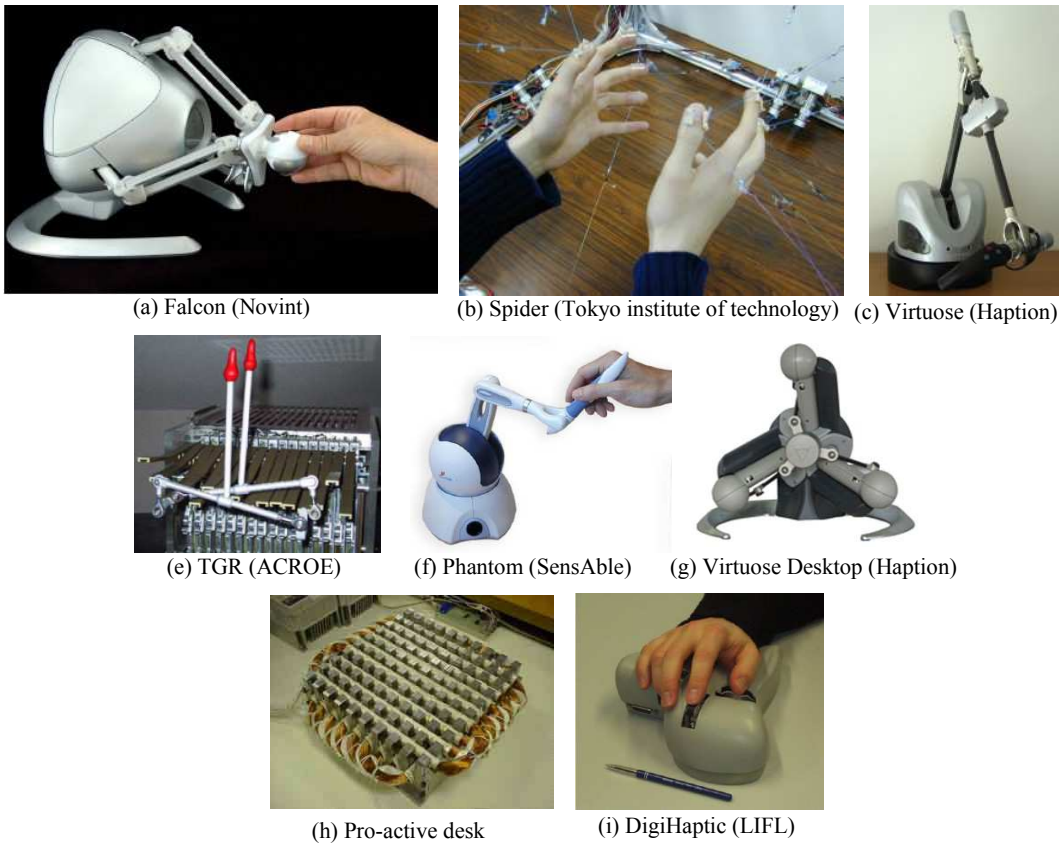
The main goal of the machine haptics research is the design, construction and usage of machines to replace or augment human touch. The machines can be roughly divided into two main categories: *tactile displays* and *haptic devices*.

Tactile displays stimulate the skin to generate the stimuli sensed by the cutaneous system introduced above (see Fig. 3.2). The mechanical displays usually utilize vibrations or small shape changes in order to simulate haptic textures or some special properties as stickiness [Yamaoka et al., 2008]. Recently, a special type of non-contact airborne ultrasound tactile displays has been presented in [Iwamoto et al., 2008]. Besides the mechanical displays, thermal tactile displays have been also developed based on thermal cues, e.g. simulating various material properties [Ho and Jones, 2007].

Haptic devices are based on positional and force sensing (see Fig. 3.3). The name is slightly misleading, as it usually denotes the machines connected to stimulation of the kinesthetic system introduced above. Usually, the devices are composed of physical components (motors, arms etc.). However, some advanced technologies have been also employed, such as magnetic levitation [Berkelman et al., 1996] or pro-active desk [Noma et al., 2003] (see (h) in Fig. 3.3).

An important criterion for the classification of the haptic devices is the number of DOF which represents the dimensionality of the haptic space ranging usually from two to six. The 3−DOF devices are perhaps the most common case, when x, y, z components of position and force are considered. This is further enhanced by adding the rotational DOFs (roll, pitch and yaw) resulting in 6−DOF devices. In [Kim et al., 2003], a 7−DOF haptic device has been reported adding 1−DOF for grasp.

Another criterion is represented by the control-paradigm of the device: *admittance* and *impedance* devices [Srinivasan and Basdogan, 1997]. The choice between these two main architectures raises some important implications in the design of the interface and the associated control loop.



(a) Falcon (Novint) (b) Spider (Tokyo institute of technology) (c) Virtuose (Haption)

(e) TGR (ACROE) (f) Phantom (SensAble) (g) Virtuose Desktop (Haption)

(h) Pro-active desk (i) DigiHaptic (LIFL)

Figure 3.3 – Kinesthetic interfaces.

Admittance devices measure the force applied by the user and constrain the position by moving the device. This kind of device can efficiently render contact surfaces by constraining the position regardless of the force applied but raises some difficulties to render "transparent" unconstrained free motion.

Impedance devices measure, on the contrary, the position of the user and produce forces. They can thus easily render free motions by not producing any force but cannot efficiently constrain the position on virtual contact surfaces.

Detailed survey concerning the haptic interfaces and devices can be found in the reference of [Hayward et al., 2004]. The real-time deformation model proposed in this thesis is intended for haptic interaction using the admittance control paradigm.

3.1.3 Computer Haptics and Haptic Rendering

The computer haptics is the third area of the haptic research dealing with paradigms, algorithms and software for the haptic interaction. It can be regarded as a step for bringing the sensation of touch into the virtual environment [Robles-De-La-Torre, 2006].

Although one can divide the computer haptics into two areas, the border between them has become quite blurred. First, it is denoted as the *haptic visualization* which represents the abstract realization of information through the use of tactile and haptic

devices [Roberts and Paneels, 2007]. The second area is the *haptic rendering*, which enables a user to touch, feel, and manipulate virtual objects through a haptic interface [Basdogan and Srinivasan, 2001].

Haptic rendering turns out to be more complex than visual rendering for two reasons.

- Haptic rendering is bilateral process, as the display cannot be separated from the manipulation. Therefore, the forces which are to be rendered must be computed in real-time during the interaction and delivered to the user via the haptic device.
- Haptic system requires signals which are refreshed *at least every millisecond*. These requirements are driven by human haptic ability to detect vibrations which peaks at about 300 Hertz, nevertheless frequencies over 1000 Hertz can be still distinguished under some conditions [Basdogan et al., 2007].

The haptic device is usually represented by a virtual object - *probe* being coupled with the haptic interaction point (HIP). During the interaction, the probe interacts with some other object in the virtual scene. A generic *haptic rendering algorithms* is then composed of two phases:

collision detection: the collision between the probe and other virtual objects must be detected with respect to the actual position of the probe;

collision response: the collision between the objects results in reacting forces, which must be computed and delivered to the haptic device[2].

It is required that the haptic rendering task is performed within the haptic loop, which runs at 1000 Hertz due to the reasons presented above. The complexity of each phase depends on the particular model, where two main classes are distinguished. Firstly, *haptic rendering of rigid objects* - a concept of proxy [Zilles and Salisbury, 1995] was introduced. The force is then rendered as a linear string between the position of the proxy object and current position of the HIP. Secondly, *haptic rendering of deformable objects* - the interaction between the probe and virtual body leads to the deformation of the either the probe, or the body or both. An example of geometrically-based haptic rendering of deformations using splines can be found in [Hua and Qin, 2002]. In the case of physically-based deformations, an example of mass-spring model used for haptic rendering of global deformation is given in [Tong et al., 2007]. Various models using finite element deformation modelling in connection with haptics are presented in the following sections.

3.1.4 Preliminary Conclusion

In the Section 1.3.1 and Section 1.3.2 of previous Chapter 1, we presented various VR applications in general and industrial evaluation applications concerning the VR - PLM integration, e.g., haptics-based assembly and interactive stress analysis in VR [Faas et al., 2007]. Here, we would like to stress the important roles of various haptic

[2]For the admittance control paradigm, the position/speed of HIP is delivered to the haptic device.

models by mentioning the following applications.

Medicine: the design and implementation of the medical simulators including haptic interfaces has been of a great interest mainly for surgical training and operation planning [Gosselin et al., 2010].

Physical and chemical simulation: haptics can significantly improve our understanding of various phenomena where force fields play an important role. An important example is given by molecular docking [Daunay et al., 2007; Daunay and Regnier, 2009]. Also, haptics was employed in the area of climate simulations in order to improve the visualization of climate data [Yannier et al., 2008].

Facilities for impaired people: haptic models can help visually impaired people, e.g. with spatial orientation in buildings [Pokluda and Sochor, 2005].

Remote Collaboration: haptics can provide another dimension for experience sharing between remote parties [Kim et al., 2004]. This concept is also explored in telesurgery [Zhou et al., 2004].

Education: the haptic e-learning can facilitate students' understanding of difficult concepts, e.g. in physics [Hamza-Lup and Sopin, 2009].

Entertainment: haptics can be used to augment the interaction in virtual worlds, such as *SecondLife* [de Pascale et al., 2008].

In the above applications, the key issue of the haptic rendering task is the high refresh rate required to keep a stable haptic interaction, which is computationally demanding. This is especially true for the goal of this thesis: real-time haptic interaction with complex industrial deformable mechanical parts, because a trade-off between the deformation accuracy and the real-time interaction performance has to be obtained. In the subsequent chapters, we will describe our two-stage method in detail to demonstrate how we resolve the trade-off issue concerning the design evaluation application investigated in this thesis. Our method combines an off-line pre-computation phase and an on-line haptic interaction phase. By employing the method, all the time-consuming computations are performed in the off-line phase and thus the real-time deformation interaction is quickly calculated. According to our simulation experiments, the two-stage method on this thesis efficiently deals with the trade-off between accuracy and speed.

3.2 Deformation Modelling for Haptic Interaction Purpose

A large amount of work has been done concerning interactive simulation of physically-based elastic objects, e.g., in computer graphics [Nealen et al., 2005]. However, only a relatively small subset of these related work has addressed the simulation requirements of force feedback concerning computer haptics. Much of the computer haptics literature on deformable models has been concerned with surgical simulations [Cotin et al., 1999;

Weiss et al., 2003]. In this thesis, we aim at interactive design validation of industrial deformable mechanical parts in the background of virtual prototyping.

In the following sections, it is not our intent to survey all the haptics work, but to mention some relevant work combining kinesthetic force feedback with elastic objects.

3.2.1 Overview

Broadly speaking, there are many techniques that are employed to model deformable objects. The survey papers by Gibson and Mirtich [Gibson and Mirtich, 1997] and Nealen et al. [Nealen et al., 2005] described various methods in detail for deformable objects modelling. As we presented in the previous chapter, two main classes of methods which are denoted as non-physically based methods and physically-based methods are clarified. The deformable models based on physically-based methods are computationally expensive, but they provide higher fidelity deformation results comparing with the models formulated by the non-physically based methods. The FEM is superior to the other physically-based methods when mechanical parts are considered for the deformation evaluation purpose, because it can better describe the behaviours of these deformable parts and enhance realistic interaction experience for real-time haptics.

Nevertheless, haptic applications require approximately 1000 Hertz for the refresh rate of haptic loop, while graphics applications require 30 Hertz to 60 Hertz [Saddik, 2007]. A trade-off between the deformation accuracy and the real-time computation speed has to be achieved concerning complex mechanical parts. This is because deformable parts derived from the FEM could lead to huge computations and thus hinder the performance of real-time interactivity. Therefore, the issue of coupling a heavy computation process with high refresh rate of the hatic loop is a challenge.

Basically, two primary categories of solutions can be identified to solve the challenge: either simplified models based on linearisation or reduction are applied, or some precomputation is employed before the real-time interaction occurs. In the following sections, we present various but well classified methods which are dedicated to handle the challenge issue discussed above.

3.2.2 Improved Methods

As described in section 2.1.1.2 of chapter 2, the major drawback of mass-spring-damper models (MSD) is the insufficient approximation of underlying material properties. Concerning the finite element models, they could lead to large systems of algebraic equations and the heavy computation loads, which could block the real-time interaction performance. In this section, we present related work either aiming at improving the accuracy of MSD models or accelerating the computational speed of finite element models.

3.2.2.1 MSD Model Accuracy Improvement

Traditional mass-spring systems are usually expressed in terms of stiffnesses for each spring, so the only way to change the behaviour of a underlying material is to vary the value of these stiffnesses. Recently, there have seen a growing interest in the research on the stiffness value identification of a mass-spring system. Two central approaches have been proposed so far to improve the simulation accuracy of MSD models.

The first approach focuses on the determination of mathematical relationships in the computation of mesh properties of MSDs based on known values or reference models. In [Gelder, 1998] stiffness values in triangulated spring meshes were computed proportional to triangle area and Young's modulus. Generic methods for particle-based systems (also termed "generalized mass-spring systems") have been suggested by [Maciel et al., 2003]. A good physical material parameters that describe materials in continuum mechanics were applied in the derived particle system of [Etzmuss et al., 2003]. Training approaches based on known reference models (e.g., the FEM models) was introduced by [Bianchi et al., 2003, 2004] to obtain a better configuration of mass-spring system parameters. In [Morris, 2008], the authors provided a pipeline for a rapid and automatic mesh preparation of deformable objects without manual interventions while taking a finite element analysis result as a ground reference.

The second approach is characterized by a series of optimization processes, which attempt to adapt the behaviour of a MSD model for different simulation scenarios. For instance, the authors of [Deussen et al., 1995] suggested to employ the simulated annealing to identify spring constants. Neural networks were regarded as an optimization method in [Nurnberger et al., 1998] for the dynamic simulation of a MSD model. And other optimization methods based on genetic algorithms for stiffness value determination were discussed in [Garat and Laugier, 1997; Louchet et al., 1995]. Concerning the coupling between a MSD model and a complex haptic virtual environment, a discretized Medial Axis Transform (MAT) method was proposed by [Corso et al., 2002] to obtain a realistic deformation simulation.

3.2.2.2 FEM Speed Improvement

In this section, we overview the proposed methods that are employed to accelerate the computational speed of finite element (FE) models. The model reduction method is preferred to improve the real-time performance of FE models. However, the related work presented in the following paragraphs focuses on the non-reduced FE models. The model reduction method will be detailed as an independent method in the subsequent section (see section 3.2.3).

Multi-resolution methods. These methods employ a hierarchical deformation method to adaptively refine the analysis respecting deformation activities of the model. Such methods exploit the fact that a problem can be solved on different scales of resolution. In the field of computer graphics, hierarchical methods have been applied to solve a variety of problems, ranging from graphical renderings [Gortler et al., 1993], geometric modellings [Gortler and Cohen, 1995], to deformable objects simulations. Terzopoulos *et al.* [Terzopoulos and Fleischer, 1988] proposed a multi-grid deformation solver on a rectangular domain. Debunne *et al.* [Debunne et al., 1999] presented an interactive deformation simulator based on an octree representation which was adaptive in both space and time (see Fig. 3.4). An extension of real-time octree representation was proposed by [Nesme et al., 2006] in which the camera position was also considered to obtain a realistic rendering effect.

The multi-resolution method is promising when haptic interfaces are introduced into these deformation simulations, because the multi-resolution representation of a complex deformable object promotes to reach a good trade-off between the real-time

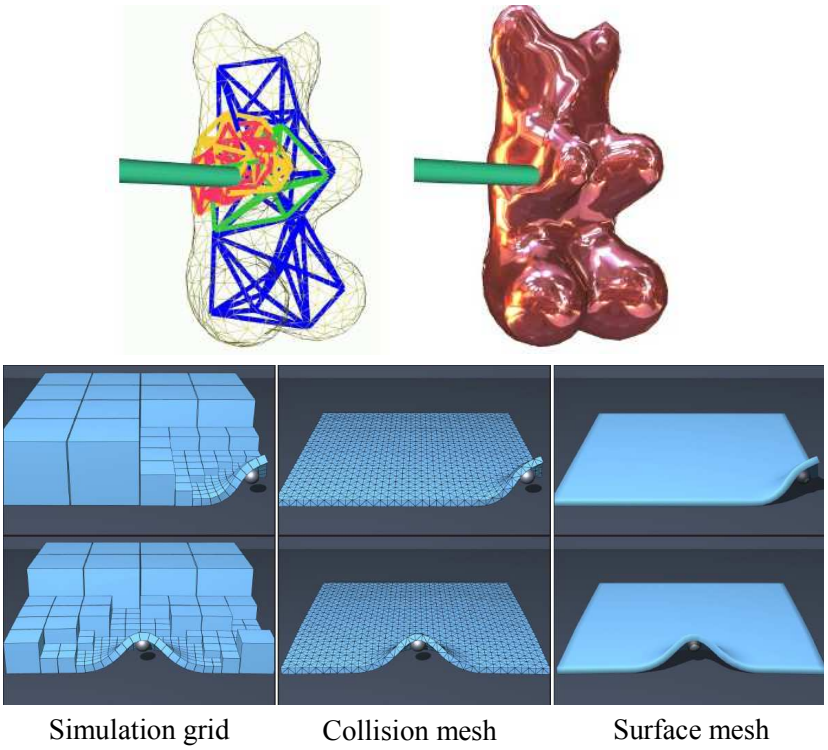


Figure 3.4 – Multi-resolution method to simulate deformable objects. Top: a non-nested multi-resolution hierarchy approach of [Debunne et al., 1999]. Bottom: a three-fold representation of a large object of [Seiler et al., 2010].

interaction performance and the deformation fidelity. For example, authors of [Koçak et al., 2009] proposed a deformation modelling method which was composed of a series of real-time adaptable regions (termed "asynchronous regions"), and each of these regions could be resolved by different solution methods with different frequencies. In order to realistically simulate small areas of the deformable surface being in contact, intermediate layers corresponding to these surface areas were introduced by [Böttcher et al., 2010]. Some variations of the multi-resolution method were presented concerning haptic contacts or collision handlings. A two-level layered deformable model was proposed in [Galoppo et al., 2007]: the low-resolution proxy was employed to accelerate the computation of collision detections and the contact force renderings, and a high-resolution tetrahedral mesh was applied to achieve highly detailed deformations. Similarly, a three-fold representation of a complex deformable object was introduced by [Seiler et al., 2010]: a coarse simulation mesh, a mid-resolution tetrahedral collision mesh, and a high-resolution surface for the visualization (see Fig. 3.4).

Concerning the mesh analysis method (see section 4.2.3) proposed in this thesis, it can be viewed as *static* multi-resolution method. By "static", we mean that the different mesh models based on variant meshing resolutions are prepared off-line. And the main task of real-time interaction is to choose a mesh model as well as the corresponding deformation space information with respect to the interaction situation. In this thesis, the real-time interaction situation is determined by the fact that which part (e.g., local areas) of the deformable object is under deformation. A major limitation of this method is that we assume the real-time interaction situations are foreseen. As it

turns out, however, the real-time constraints, e.g., the contact conditions, are normally time-varying and it is not possible to predict in an off-line phase.

GPU-Accelerated FEM. The introduction of the GPU[3] into the process of deformation computations provides an efficient alternative for massive data-parallel computations. Besides traditional graphics rendering, the GPU is capable for a variety of data-intensive or time-critical applications. Deformable objects formulated by the FEM result in a high computational complexity, and the exploration of GPU's parallel matrix and vector computational power provides possibilities to accelerate the computational speed of such models. A very general solution of applying the GPU algorithms to the linear FEM deformations was presented by [Liu et al., 2008], in which their implementation was based on CUDA[4]. In contrast, a fully non-linear FEM simulator employing CUDA was efficiently implemented by [Comas et al., 2008] in a framework SOFA[5], and moreover their application considered viscoelastic and anisotropic features of the mechanical response of the material due to the computational power of the GPU and thus enhanced the fidelity of tissue deformations for haptic interactions. Besides, the potential enhancement of applying GPU for haptic interaction with real-time deformations was studied by [Georgii and Westermann, 2005; de Pascale et al., 2004; Shi and Payandeh, 2008].

There are other methods that are proposed to speed up the real-time deformable simulation of finite element models. For example, a mass-lumped method was introduced by [Zhuang and Canny, 1999] by diagonalizing the mass matrix and the damping matrix in order to solve a non-linear FEM formulation for enabling real-time haptic interactions.

By contrast, we do not take the computational advantage of the hardware in our two-stage method. That is, our experimental results in terms of the real-time haptic interaction performance are resulted from the implementation on CPU. However, the parallel computational feature of GPU could contribute to the haptic interaction with complex industrial mock-ups in terms of speeding up the real-time deformation and the graphical rendering.

3.2.3 Dimensional Model Reduction Method

The *Dimensional Model Reduction* method has been explored extensively in the fields of control theory, electrical circuit simulation, computational electromagnetics and microelectromechanical systems [Bai and Li, 2005]. This method also appears in literature under the names of *Principal Orthogonal Directions Method*, or *Sub-space Integration Method*, and it has a long history in the engineering and applied mathematics literature [Lumley, 1967]. The common theme of this method is to project the original state-space on a low-dimensional sub-space to reach a much smaller system which manifests similar properties comparing with the original system. Applications of applying the model reduction method to fluid simulations can be found, for example, in [Ko et al., 2000; Treuille et al., 2006].

[3] GPU: Graphics Processing Unit.
[4] www.developer.nvidia.com/object/cuda
[5] www.sofa-framework.org

The key idea of applying the dimensional model reduction method for deformation simulations is to simplify a dynamical system formulated by a series of differential equations. By this method, the dimensionality of the original finite element problem can be considerably reduced, yielding a system consisting of fewer differential equations and fewer unknown variables. These equations can be solved much more quickly in real-time with some accuracy lost to the solution. The key issue of the model reduction method for deformation simulation is to choose a sound deformation sub-space in which the deformation can be admirably approximated. And normally the process of generating such deformation space is based on carefully designed pre-computations. Recently, a statistical approach for deformation basis generation concerning finite element models was presented by Krysl *et al.* [Krysl et al., 2001] wherein a full-degree of freedom system was first simulated, and then standard Principal Component Analysis (PCA) was applied on the recorded deformation data to obtain a typical deformation basis. The most recent work is in [Barbič and James, 2005], where they introduced two methods to generate reduced deformation basis: an interactive method based on the user feedback and another automatic way based on the linear modal analysis.

A somewhat different model reduction method was devised by James and Fatahalian [James and Fatahalian, 2003]: they entirely pre-computed collision, contact and dynamics as well as parameterized global illumination of deformable objects by limiting the range of interactions to a finite set of user impulses. The dynamic responses, termed impulse response functions (IRFs), each a chronological set of generalized deformation vectors, were dimensionality reduced using PCA. At run-time, IRFs were blended based on user interactions. An Impulse Response Deformation Model (IRDM) was proposed in [Tagawa et al., 2006] by simulating the surface mesh nodes' response of a real object.

To summary, the employment of the model reduction method in the field of computer graphics aims to obtain a plausible deformation result which is normally lack of physics features. However, things are different when the model reduction method is applied in virtual prototyping applications for purpose of the design evaluation of industrial mechanical parts in a VR environment. As we explained in section 1.4.2, virtual prototyping applications are model-centred while VR application are user-centred. These two antagonist requirements lead to a trade-off issue between the deformation accuracy required by the virtual prototyping application and the interaction performance required by the introduction of users into VR applications. More particularly, a good choice of deformation space to represent the behaviour of real-time deformable objects is of first importance. Data-based methods and the modal analysis share the same aim at the computation of deformation spaces in an off-line phase. However, we present the difference of these two methods in the following sections.

3.2.4 Data-based Methods

As we discussed in the previous sections, deformable objects formulated by the FEM are usually employed for generating high fidelity simulation results. However, such models lead to a heavy computation load in real-time, because algorithms of the FEM need to build a mathematically explicit model (see Equation 2.15). Consequently, the parametric deformable models tend to largely decrease the interactive rate, and thus raise difficulties for real-time haptic interactions with such explicit FEM models.

On the contrary, the data-based deformation modelling method does not require an explicit deformation formulation. The information about the underl ying material property is not explicitly required. Therefore this method can be applied on linear, non-linear or any other material types. The idea behind the data-based method is to collect enough data to describe the behaviour of deformable objects. Here, the word "enough" means that it is not a full data collection because they are finite but it is enough to achieve deformation results very close to parametric deformable models. Moreover, the data-based method is appropriate for haptic interaction simulations [Tagawa et al., 2008] as the parametric techniques are still not appropriate for real-time complex non-linear simulations.

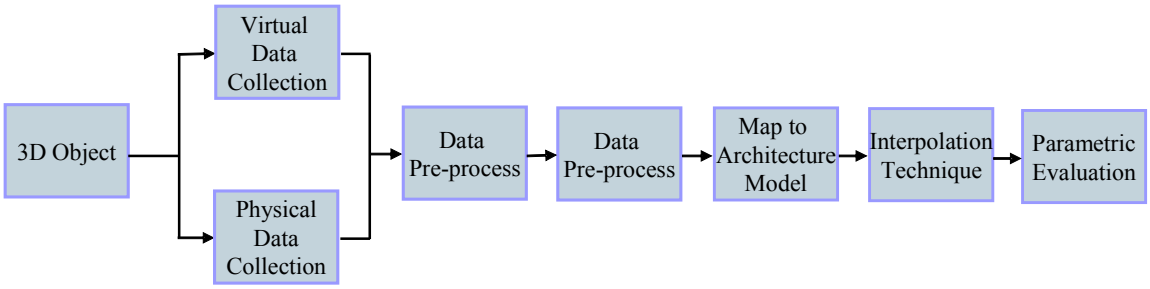


Figure 3.5 – The procedures of data-based modelling approach for haptic simulation of deformable models [Wael et al., 2010a].

As discussed in [Wael et al., 2010b], most data-based deformation modelling methods share common steps, as illustrated in Fig. 3.5.

- *Data Gathering.* This step includes the employment of hardware equipment [Pai et al., 2001] or simulation software (as discussed in section 2.4) to generate the deformation data. In addition, different data collection strategies were proposed to achieve data with different purposes. For instance, all possible data was considered [Fong, 2009], random amount of data by a neural network [Morooka et al., 2008] or a planned selection on surface meshing points [Wael et al., 2010a].

- *Pre-processing.* The collected data might not be ready to use due to the noises or the high dimensionality. In this step, data are processed and filtered according to the deformation modelling requirements. As discussed above, the PCA is generally employed to reduce the data dimensionality, yielding a real-time costless deformation response model.

- *Model Architecture Selection.* Here the collected data are mapped to a response model which decides how the data will be manipulated in real time. Usually the architecture is an interpolating mechanism [Peterlík and Matyska, 2007], such as Artificial Neural Network (ANN) [Cretu and Petriu, 2006] or Radial Basis Functions (RBF) [Hover et al., 2009].

- *Learning.* This step aims at building an real-time system that can simulate the phenomena based on the previous observations. And the system should manifest the capacities of simulating the recorded deformation behaviour as well as the new on-line behaviours.

- *Parametric Evaluation.* This step includes the processes of checking the accuracy, feasibility, and completeness of the deformation results.

In our two-stage method, we pre-compute deformation spaces based on anticipated industrial evaluation scenarios. However, our method is not a data-based deformation modelling method, because during the pre-computation phase we fully respect the material properties and we do not regard the data-collection procedures as a "black-box".

3.2.5 Modal Analysis Method

Each solid object possesses natural "eigendeformation" shapes (called modes), with a certain natural frequency for each shape. These shapes depend on the object geometry, boundary conditions (such as what parts of the object are fixed), and material properties. In many applications, the lowest natural frequencies and their corresponding shapes provide a reasonable representation to describe the behaviour of a deformable object. The deformation approximation by choosing certain modes is convergent in the sense that if more modes are added into a deformation basis, and thus a better accuracy is obtained. If all modes are employed, one obtains the same accuracy as under a full model.

The modal analysis method was introduced to Computer Graphics in 1989, but only relatively few significant contributions have been presented since then. The first work was Stam's application of applying the modal analysis to the simulation of tree branches subjected to turbulence [Stam, 1997]. James and Pai [James and Pai, 2002] mapped real-time dynamics to graphics hardware: in a pre-computation stage they built a so-called Dynamic Response Texture (DyRT), where mode shapes and other quantities were stored. At run-time, the modal coordinates $\mathbf{q}$ (called reduced coordinate) were computed from rigid bone transforms or external excitations.

Enforcing direct manipulation and collision constraints is straightforward with node positions in Euclidean space. However, applying these operations in a modal space can be unintuitive. Hauser *et al.* [Hauser et al., 2003] provided a solution, in which generalized forces are computed for constrained nodes based on the modal basis. Since the force computation involves evaluating a pseudo-inverse applying the singular value decomposition (SVD), only few constraints (up to ten in their examples) can be applied in a real-time simulation environment. Furthermore, they coupled the deformable model with a rigid body reference frame to simulate dynamics, collisions and frictions.

In the above methods, a linear Cauchy strain model (see Equation 2.9 and 2.10) is employed to obtain a constant stiffness matrix K, resulting in artifacts for large rotational deformations away from the rest shape. To suppress these artifacts, Choi and Ko [Choi and Ko, 2005] identified per-node rotations and extended the basic modal analysis formulation to accommodate these rotations, similar in spirit to the warped stiffness approach [Müller et al., 2002]. Although their modal warping procedure was not guaranteed to perform well for large deformations, their results illustrated visually convincing improvements over the standard linear model.

Barbič and James [Barbič and James, 2005] introduced a different approach: instead of employing Cauchy's linearized strain they employed full quadratic Green strain throughout the entire computation, thereby necessitating the solution of a non-linear version of Equation 2.16 per time step. The otherwise computationally burdening implicit Newmark integration was greatly accelerated by carrying it out in reduced coordinates (also called sub-space integration), and thereafter performing the matrix multiplication of Equation 2.20 in the fragment shader on graphics hardware. For

the generation of an appropriate reduced deformation basis **U**, which contains sufficient non-linear deformation, they provided a fully automatic method based on modal derivatives, and also illustrated how user-defined (force) sketches could assist a full unreduced off-line static solver for pre-computing a suitable **U**.

3.2.6 Preliminary Conclusion and Overview of Our Method

The model reduction method is more attractive for the real-time haptic interaction with deformation simulations, as most of the computational overhead of modelling deformable objects is performed during the pre-computation process. Therefore the computation power saved during real-time interactions can then be dedicated to model other complex deformable behaviours, like visco-elasticity, and non-linearity. However, the model reduction technique based on pre-computations raises several difficulties. The first one is to compress the amount of data collected during the pre-computation process, because the reduction of the collected data is essential for a quick access to them in real-time interactions[Tagawa et al., 2009; Dulong et al., 2007]. It is true that the number of possible interactions is usually large, since enough pre-computations should be done before real-time potential interactions occur. The size of the collected data depends heavily on the meshing resolution of a deformable object and the coverage of required deformation behaviours. This leads to the second difficulty: how to design an efficient data collection strategies. Without planning the method of data collections, the pre-computation process will be time-consuming. In [Wael et al., 2010a], the authors proposed an adaptive method for the deformation basis generation employing a specific collection planning technique based on object surface segmentations. Another difficulty is to generalize the pre-computed data to different scenarios. In other words, the pre-computed data should possess the capacity to predict new scenarios that does not exist in the collected data, for example by implementing the interpolation technique [Peterlík and Matyska, 2007], or the extrapolation technique [Picinbono and Lombardo, 1999].

By contrast, regarding our goal application and industrial design evaluation requirements, the two-stage method proposed in this thesis efficiently avoids the aforesaid difficulties raised from the model reduction method. First, during the pre-computation phase, our method computes a linear deformation space based on the modal analysis, similar with the proposition in [Hauser et al., 2003]. Since our pre-computation step is not a random data collection process, we obtained modal deformation vectors and the corresponding modal parameters in a mathematically rigorous way. Moreover our experiments show that this process is not time-consuming and the data storage is small. Second, our pre-defined global and local scenarios provide an opportunity to plan the pre-computation process efficiently. Regarding these scenarios, we propose a mesh analysis technique, with an accuracy enhancement, which is dedicated for these anticipated scenarios, and this is one difference compared with the work in [Hauser et al., 2003]. Third, considering the different refresh rates of the haptic rendering and the visual rendering, we propose to divide deformation computation process into two separate modules: a haptic rendering module and a visualization module, which are actually implemented on different threads. Our haptic rendering process can be quickly refreshed at more than 1000 Hertz by extracting a sub-matrix from the pre-computed data matrix. However, the update rate of visualization process is lower and it heavily

depends on the meshing densities.

3.3 Chapter Summary

In this chapter, we presented the related work in terms of the employment of haptics in virtual reality applications and physically-based haptic interaction with deformable objects. To conclude, we intend to emphasize the following two points. On one hand, the model reduction method is attractive and more suitable for real-time haptic interaction with industrial deformable mechanical parts. On the other hand, the two-stage method proposed in the thesis efficiently resolves the difficulties raised from the model reduction method, e.g., in terms of the time for pre-computations, the storage of collected data and the capacities to generalize collected data to different scenarios.

In the next chapter, we present the two-stage deformation modelling method in detail.

Our Haptic Deformation Model

4

Contents

In the previous chapter, we presented an overview of the related work on physically-based deformation modelling methods and the crucial role of haptic interfaces for providing realistic sensations to an operator. In this chapter, we present a detailed description of our haptic deformation model, which is a computational model connecting the deformation modelling on one hand and the haptic interfaces on the other. The main goal of this chapter is to describe the data flow of the haptic interaction with deformable objects.

4.1 Overview of Our Haptic Model

This section provides a detail description of the computation model of the haptic interaction with deformable objects. The main goal of this proposal is to identify the key components of the haptic model employed in this thesis.

It is important to point out that the haptic model would be modified to adapt the case when rigid-body motions are considered. In such a case, our haptic model, particularly, refers to the process of the behaviour modelling of deformable objects on one hand and the haptic interactions and the data flow on the other. By "behaviour

modelling", we mean the modelling process of the combination of a small deformation and a large rigid-body motion of a deformable and moveable object.

The first step towards a definition of the haptic deformation model is composed by illustrating a connection between the finite element modelling on one side and the haptic interface on the other. Since the connection between the two parts is represented by a data flow, the components representing the inputs and outputs are identified first on the both side (see Fig. 4.1). In the following subsections, we discussed the haptic model from three aspects: the key components of FE modelling, the rigid-body motion modelling, the admittance-controlled haptic interfaces and the data flow in the haptic model.

4.1.1 Components of FE Modelling

In the following description, the terms that appears in Fig. 4.1 are displayed by a typeset in italic. The formulation of a boundary-value problem consists of two steps. In the first step, the material property must be specified by selecting a material law and the corresponding material coefficients. In the second step, the domain of the deformable body together with boundary conditions must be defined.

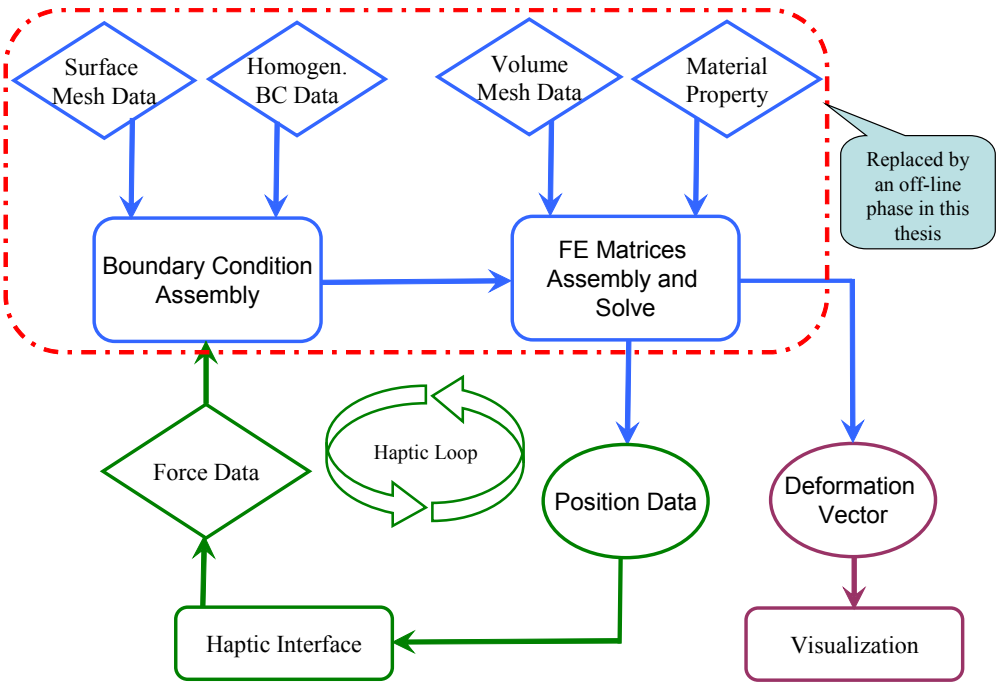


Figure 4.1 – Components and relations between them in the haptic deformation model.

In this thesis, as the finite element method is employed for the deformation modelling, the elastic problem is formulated over a geometric domain represented by the finite element mesh which consists of two components as follows. The *volume mesh data* are applied for the procedure of assembling the FE matrices, while the *surface mesh data* are used primarily for the definition of the boundary conditions. When employing the FEM, the boundary conditions (BC) are described by the prescribed values of the variables (e.g., displacement) over some parts of a deformable body surface. First, the *homogeneous BC data* are represented by the areas of surface in which the prescribed values of the variables are set to zero (e.g., zero displacement). Second, the heteroge-

neous BC data are represented by non-zero values of the variables prescribed to some areas on the body surface. Another component in the finite element modelling is represented by the *applied force* (e.g., gravitation or a surface traction). All the components required for FE modelling are illustrated in the frame of red dot-dash-line. In this thesis, this part is replaced by an off-line phase. The model exportation, meshing, and the BC definition are implemented manually, which will be presented in the subsequent sections concerning the data flow during our pre-computations.

4.1.2 Rigid-body Motion Modelling

During the real-time interaction process, the rigid-body motion of a deformable and moveable object is computed by solving the following classical equations,

$$M\frac{dv}{dt} = \sum f_{ext} = \mathbf{F} \tag{4.1}$$

$$w \times (Iw) + I\frac{dw}{dt} = \sum \tau_{ext} \tag{4.2}$$

where, M is the total mass of the elastic object, I is inertia tensor, v and w are respectively velocity and angular velocity of a rigid object. $\mathbf{F}$ and τ_{ext} are external force and external torque that are applied by users. In the implementation, Euler's method was used to numerically compute the solution as those in [Tagawa et al., 2008].

The combination of the components of rigid-body motion and deformation is detailed in section 5.3.3.

4.1.3 Admittance-controlled Haptic Interfaces

After specifying the main components of deformation modelling, we now focus on the haptic interface. As introduced in section 3.1.2, the admittance-controlled haptic devices describe a haptic interface that measure the force applied by a n operator and constraint the position by moving the device. Therefore, the only output of provided by the haptic interface is represented by a vector of *force data*, which contains for example the actual x, y and z components of the forces or torques. On the other side, the only input that is delivered to the device is represented by a vector of *position data* which contains the x, y and z position of a virtual probe representing the device tip in 3D space and additionally, roll, pitch, and yaw of the device arm in addition for the case of 6-DOF devices.

4.1.4 Data Flow in Our Haptic Model

After indicating the inputs and outputs of both the FE modelling and the haptic interface, the connection between the two parts is established to define the main loop of the haptic model. First, the force data specified as the output of the haptic interface can be associated with the BC being taken as the input of the FE modelling. Equally, the output position data of the virtual probe computed by the FE modelling component can be identified and delivered to the haptic interface. The data flow inside the haptic model is then as follows:

1. the force data are acquired from the haptic device based on the admittance-control theme;
2. the boundary conditions are assembled by using the mesh surface data;
3. the FE system is assembled over the mesh volume by employing the material data and boundary conditions;
4. the response of the FE system is calculated and the deformation vector is achieved;
5. the position data are retrieved from the solution and delivered to the haptic device and the deformation of the whole body is visualized.

The five-step procedure informally establishes one iteration of the closed haptic interaction loop which is the main loop of the haptic model presented in Fig. 4.1.

It is necessary to point out the following two points. First, the haptic interaction loop introduced above is designed as *force-driven*, which means that the computations within the FE modelling are initiated by the change of the force data provided by the haptic devices piloted by a user. The position of the device tip are also incorporated into the modelling to determine the collision between the probe and the deformable object. In this thesis, we employ the position data of the tip to identify the surface vertex which is *snapping* with the probe. Second, the step **2** and the step **3** are not explicitly implemented in this thesis, as these two steps are included in our pre-computation phase. The response of the FE system and the retrial of the position data to be delivered to the haptic interface are computed in our real-time phase. The computational details of both phases will be presented in the following chapter.

4.2 Description of Modal Deformation Space

4.2.1 Overview

In the previous section, we presented an overview of our haptic model in terms of the components and the data flow within the model. In this section, we focus on the description of the deformation space.

Here, we describe the definition of the parameters of a deformation space for the real-time haptic interactions. Let Ω be the domain of a deformable body which is discretized by a finite element mesh determined by a set N of the nodes and the set E of the volumetric elements (3D convexes). Each node $n \in N$ is given by a vector $\mathbf{x}_n = [x_n, y_n, z_n]^T$ of its coordinates and each element $e \in E$ is given by the list of its nodes. Moreover, the set of surface nodes is denoted as N_s and the set of surface element (2D faces) is termed E_s. In this thesis, the simulation mesh is composed of the linear tetrahedral elements. Thus each 3D element is represented as a tetrahedron determined by four nodes $\langle k, l, m, n\rangle$, and each surface element is represented by a triangle of three node $\langle k, l, m\rangle$. The external applied forces are represented by a vector of force components $F = [F_x, F_y, F_z]$. Finally, the surface nodes, for which the boundary conditions are prescribed is denoted as $B \subset N_s$.

We assume that all the parameters introduced above remain unchanged during real-time interactions, e.g., no modification of the material data, mesh data and homogeneous BCs occurs, as the topological changes of the deformable parts during real-

time interactions are not considered. Several interactive surgical training applications [Delingette et al., 1999; André and Delingette, 2008] took the topology modifications into account.

The main contribution of the thesis is not the propositions of method to generate deformation spaces. From another perspective, we focus on the employment of linear modal deformation space for the purpose of the design evaluation of industrial deformable parts. One of the main contribution of the thesis is that we propose an off-line mesh analysis method based on anticipated scenarios (see Fig. 1.9 and Fig. 1.10). And the method is dedicated to the pre-computations of modal deformation spaces which are determined by the design evaluation requirements, which will be described in section 4.2.3. However, we first present the description of the modal deformation space, as this is the main component of our haptic model applied in the thesis.

4.2.2 Modal Deformation Space

In section 2.3.3 we described the formulations of linear modal analysis in terms of the deduction of the generalized eigenvalue problem and the advantages of applying the modal analysis method to deal with time-critical interactive deformation applications.

Although decoupling equation 2.16 and then solving each of the resulting components provides significant benefits, we can obtain additional benefits by examining whether or not each of the these components is necessary for the real-time deformation interactions. In particular we can discard modes that will have no significant effect on the simulations we wish to model. On one hand, if the eigenvalue λ_i associated with a particular mode is large, then the force required to to excite a discernible displacement of that mode would also be large. On the other hand, the imaginary part of ω_i determines the frequency that a mode will vibrate at. Modes that vibrate at more than half the display's frame rate will cause temporal aliasing [Hauser et al., 2003]. Therefore, removing modes that are too stiff and/or too high frequency to be observed will not change the appearance of the resulting simulation, but removing them will greatly reduce the simulation's cost.

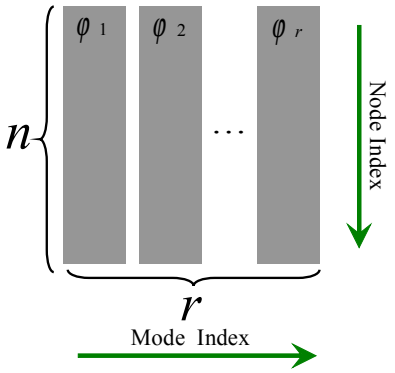


Figure 4.2 – Illustration of modal deformation data storage (r modes and n nodes).

In this thesis, we pre-compute the linear modal deformation space and store it in the way as illustrated in Fig. 4.2: deformation data storage of a deformable part with n mesh nodes and r modes chosen. In particular, such storage forms the *modal matrix* (or termed *modal displacement matrix*) Φ as presented in section 2.3.3.

4.2.3 Mesh Analysis Method based on Anticipated Scenarios

During off-line pre-computations, it is not necessary to mesh the object by applying a unified meshing resolution on a complex deformable object for real-time haptic interactions, as the deformation concentrates in a local region when an operator interacts with local ribs or stiffeners for the purpose of design evaluations. Regarding this point, we propose a mesh analysis method dedicated to such local deformation verification scenarios. This method can be viewed as an attempt towards deformation accuracy enhancements by decoupling a deformable object into separate anticipated Volume Of Interests (VOI). Moreover, another advantage of this method lies in the fact that most of the computational overhead of mesh preparations is incurred off-line, and therefore the deformations of the VOIs in the anticipated scenarios can be computed based on a denser mesh, while fulfilling the goal of reducing the real-time computational task.

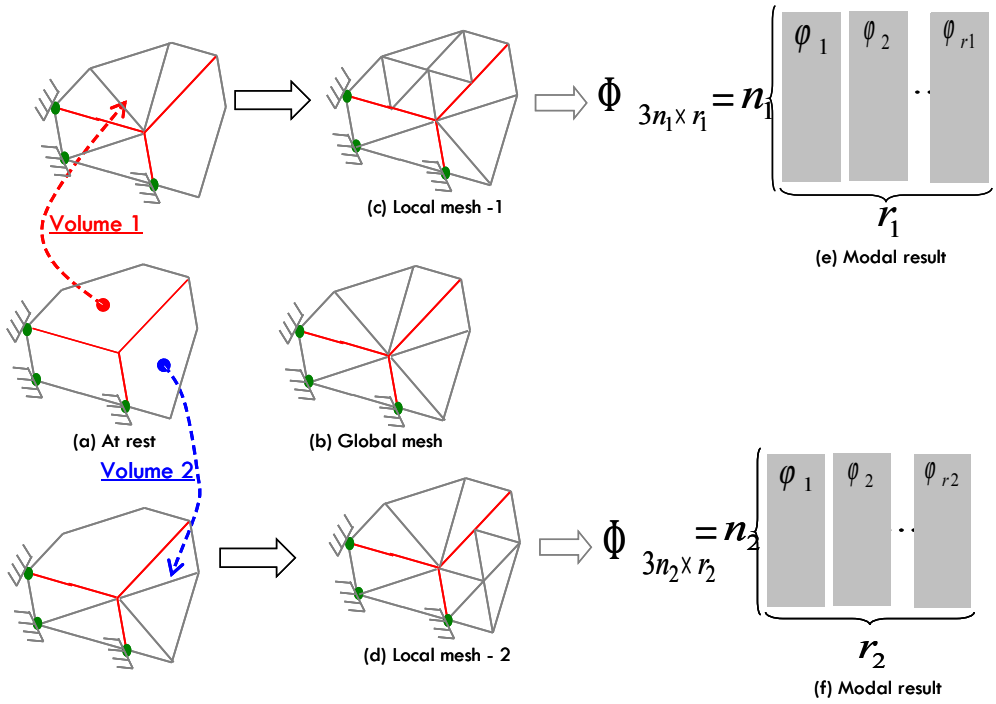


Figure 4.3 – 2D demonstration of the mesh analysis technique based on anticipated scenarios (e.g., local volume 1 and 2) – a coarse mesh is applied to an entire body. Several specified areas in different scenarios is meshed using different meshing density, (c) and (d).

Figure 4.3 illustrates the mesh analysis method in a 2D case for the purpose of simplicity. However, we consider the application of this method in a 3D case in the thesis, where the specified areas (e.g., darkcoloured areas) defined in the 2D case are replaced by a VOI. Specifically, our mesh analysis method works as follows. First, a coarse mesh is applied to an entire deformable object. We suppose that there are several specified areas in the coarse mesh, which are resulted from pre-defined scenarios (e.g., local volume 1 and 2 shown in Fig. 4.3). Second, we apply a denser mesh for each of these local volumes, while the other areas with a lower meshing density or keeping the coarse mesh. At last, modal deformation spaces (e.g., Φ^{3n_1,r_1}, Φ^{3n_2,r_2}) are stored separately with respect to different meshing qualities.

4.2.3.1 Comparison with adaptive meshing method

Our mesh analysis method considers the need for accuracy enhancements respecting different anticipated scenarios. It is true that a standard finite element method to achieve accuracy enhancement is the *adaptive meshing method* [Debunne et al., 2001]. However, the adaptive meshing method are not suitable for the purpose of real-time interactions because they are CPU intensive for simulations of 3D deformable objects mainly due to the time-consuming on-line re-meshing procedures.

In our method, by contrast, most of the overhead of mesh preparations is incurred during an off-line phase by decoupling a deformable object into separate VOIs, so that the regions in the anticipated scenarios can be computed accurately in real-time, while fulfilling the goal of reducing the on-line computational costs. The essence of our mesh analysis method is to pre-compute several mesh representation models of a single CAD object, and during real-time interactions, we choose the different representation of the same object according to the touch position of the mesh node.

4.2.4 Switch Scheme Among Modal Deformation Spaces

In the previous subsections, we presented in detail of the storage of the modal deformation data and the mesh analysis method for generating different modal deformation spaces, both of which concern the deformation described by one certain modal deformation space. In this subsection, we introduce a switch scheme allowing a real-time switch (transition) among different modal spaces with respect to the interaction contexts.

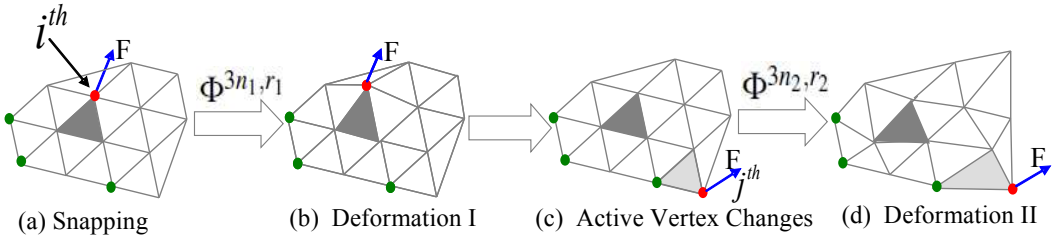


Figure 4.4 – Our automatic switch scheme for choosing modal spaces which are pre-computed off-line based on anticipated scenarios. When the index of active vertex change, the corresponding modal spaces are switched

Here, we assume that the single point interaction model is established and the details of the interaction models will be presented in section 4.3. During real-time haptic interactions, we propose an automatic switch scheme among different modal spaces which are precomputed off-line based on mesh analysis method discussed above. In practice, the switch scheme works by tracing the index of the vertex "snapped" by the HIP[1]. Specifically, Figure 4.4 illustrates the scheme in a 2D case for the purpose of simplicity. When the index (e.g., i^{th}) of the active vertex "snapped" by the HIP is located in the area surrounded by bP1, bP2 and bP3, the corresponding modal sub-space Φ^{3n_1,r_1} associated with "Local volume 1" (shown in Fig. 4.3) is chosen for real-time deformation computations. At anyone time, just one modal sub-space is "activate". If the HIP moves towards another VOI defined in the anticipated scenarios, e.g., from vertex i^{th} to vertex j^{th}, the modal sub-space is switch to Φ^{3n_2,r_2}, as vertex j^{th} is located in the area surrounded by bQ1, bQ2 and bQ3.

[1]HIP: Haptic Interaction Point.

4.3 Point-like Interaction Model

The point-like or point-based haptic interactions are commonly employed in the haptic literature for rigid surface models [Ho et al., 1999]. As stated in [James and Pai, 2005], unlike the rigid counterparts, a special care should be taken with elastic models to define *finite contacts areas* for point-like interactions since point-like contacts defined only as single-vertex or nearest neighbour lead to mesh related artifacts. However, we do not consider the definition of the small contact area during real-time haptic interactions as we explain in section 1.4.4 that global deformations are concerned in this thesis. And according to our interaction experiments, there is no observed deformation distortions resulting from the single point interaction.

In the following section, the term "single point" means a *single surface vertex* which is "snapped" by the virtual representation of the end-effector of a haptic device.

4.3.1 Single Point Interaction Model

During real-time interactions, a haptic device is represented by a single point in 3D space, called HIP, the same abbreviation as we introduced in the previous section. Figure 4.5 illustrates three interaction modes. However, we focus on single (multiple) point(s) interactions (as shown in (a) and (b)) in this thesis. A probe interaction (as shown in (c)) with complicated or irregular shape is not the scope of this thesis, as the mapping between the position of the probe and the non-homogeneous boundary conditions is not straightforward for a probe interaction.

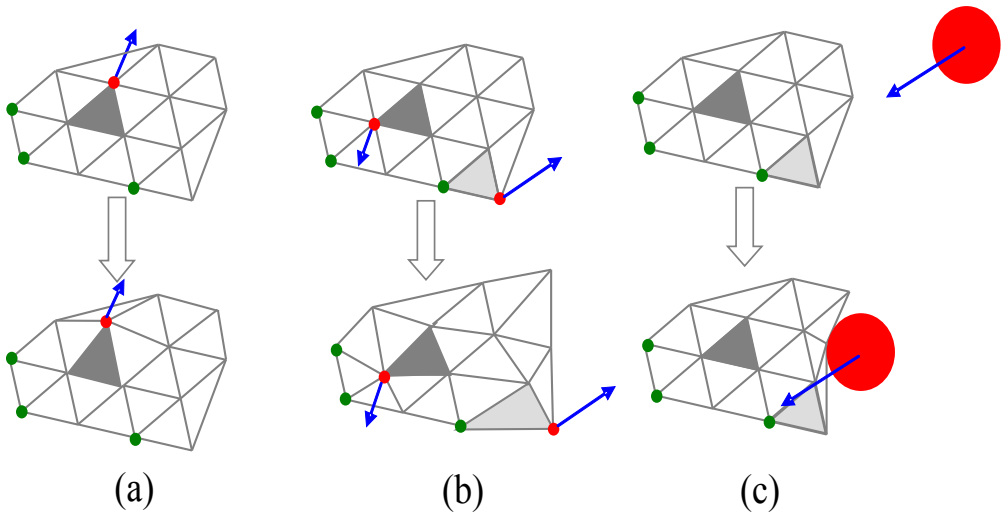


Figure 4.5 – Real-time haptic interaction modes: (a) single point interaction; (b) multiple point interaction; (c) probe interaction. Red points are active vertices "snapped" by a (two) HIP(s), while green ones are rigidly fixed. Darkcoloured regions are correspondent with those shown in Fig. 4.3.

4.3.2 Our Interaction Steps

After the off-line pre-computation process is completed, the real-time interaction starts when the mesh model and modal data are loaded. We follow the following interaction steps.

1. We move the HIP towards the deformable object.
2. If the HIP collides with the deformable object, it "*snaps*" to a vertex of the FE mesh. Without loss of generality, we assume that the HIP "snaps" to the closest vertex of the mesh which then becomes an *active vertex*. And the *index* of this active vertex is recorded. Multiple active vertices are considered if there are at least two HIPs (piloted by e.g., two haptic devices) each being attached to one active vertex.
3. Based on the admittance control[2] of a haptic device, we compute the deformation of the entire object and the speed and position of the *active vertices*. The former is used to update the visualization thread, while the latter is used to update the haptic rendering thread. The processes of these two threads are described in details in the subsequent sections.

It is necessary to point out that our real-time switch scheme among different modal spaces and our on-line division scheme (see section 5.3.2) are based on the index of the active vertex (e.g., vertex i^{th}).

4.3.3 Multiple Points Interaction Model

A two-points interaction mode is illustrated in Fig. 4.5 (*b*). The two red points are associated with the virtual representation of the tip of the haptic devices. During real-time interactions, the user can manipulate one virtual deformable object using both hands, which is correspondent to the global design evaluation scenario presented in section 1.4.2. Figure 4.6 illustrates the interaction data flow concerning a two-points interaction model and two HIPs are created and each is correspondent with the virtual representation of the tip of a haptic device. Moreover, such a paradigm is suitable for two users (the sketches of the virtual-man in Fig. 4.6) interacting with the same deformable object. The data flow of connecting two haptic devices with deformation simulator will be discussed in detail in chapter 5.

4.3.4 Comment on the HIP

In this thesis, we do not consider the tool use for the interaction with a deformable object. By contrast, a single point is taken into account to represent a HIP which is piloted by the movement of the end-effector of a haptic interface. When the HIP "snaps" with a mesh vertex, the real-time admittance-based haptic interaction starts. For industrial applications, such a interaction paradigm can be accepted, because we do not mean to perceive the surface property (e.g., textures, frictions) of a CAD model.

[2] Admittance haptic device accepts positions and return forces [Adams and Hannaford, 1999].

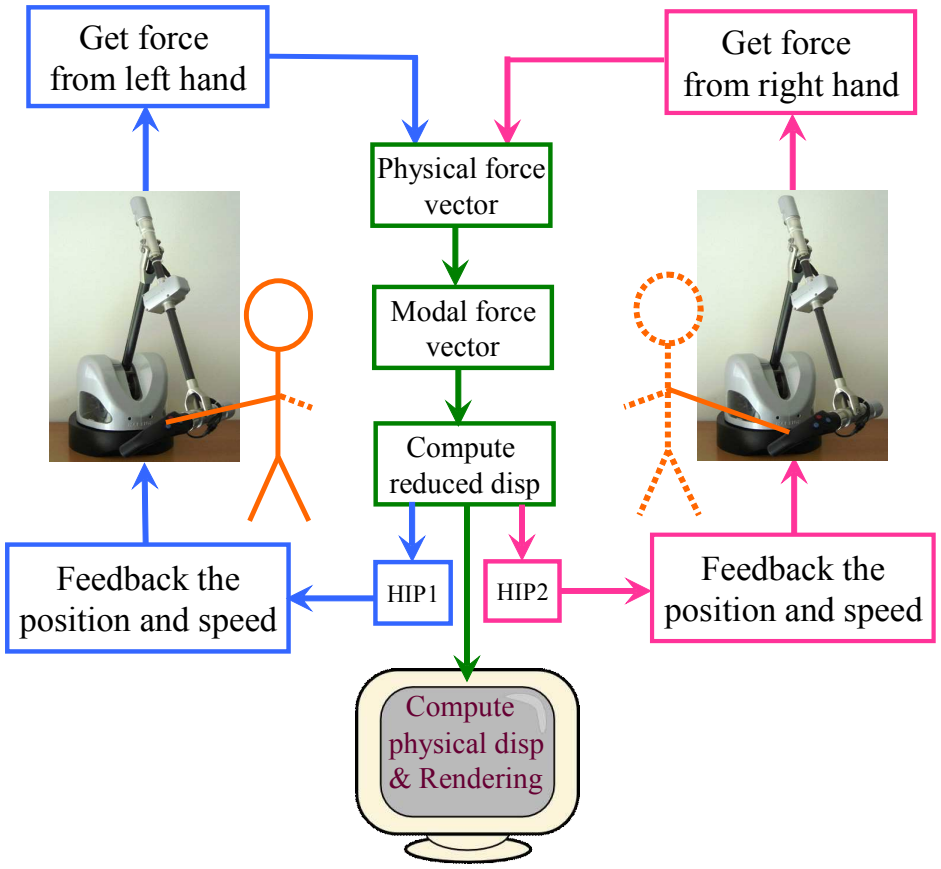


Figure 4.6 – Interaction data flow concerning a two-point interaction model with two haptic devices.

Based on the requirements of the industrial project, the stiffness perception and verification of local volume structure is more industrially crucial. Therefore, we did not handle the scenario of collision detections, for example, a single point slipping along the surface of a deformable object.

It is true that the point-like haptic interaction paradigm is a practical mode as there is no extra computation load required in terms of the collision detection task, either between a rigid tool with a deformable model [Colgate et al., 1995] or between a deformable tool with a deformable model [Laycock and Day, 2004]. This research direction would be an interesting part of the future work.

4.4 Chapter Summary

In this chapter, we presented our haptic deformation model from two aspects: the main components of deformation modelling and the haptic interaction mode. And the data flow in the haptic deformation model was introduced based on the admittance haptic control scheme (accepting positions and returning forces). For real-time deformation interactions, we focused on the single point interaction model and the direct extension to a two-point interaction paradigm.

In the next chapter, we will present the real-time simulation with off-line and on-line computations, which rely on the haptic deformation model of this chapter.

Real-time Simulation with Off-line and On-line Computations 5

Contents

As we discussed in the previous chapters, the physically-based real-time haptic interaction with deformable objects formulated by the FEM and the elasticity theory lead to a huge deformation system and thus the process to the solution is computationally expensive which hinders the real-time interactivity. One of the main contribution of the thesis is that we propose a two-stage deformation modelling method for haptic interaction purpose by combining an off-line pre-computation phase and an on-line deformation interaction phase. By employing this method, most of the heavy computational task is carried out off-line and consequently, we obtain a costless deformation response model which ensures a stable real-time interaction experience.

In this chapter, we present the detailed description of the real-time deformation simulation with off-line and on-line computations.

5.1 Overview of Our Two-stage Method

In the previous chapter, we presented our haptic deformation model from two aspect: the description of deformation modelling components and the real-time interaction steps based on single point-like interaction model (see section 4.3). In particularly, our off-line pre-computations aim to handle those deformation modelling components by proposing a feasible flow starting from an original digital model to a physical model represented by deformation spaces and the corresponding constraints. And our on-line deformation interaction phase dedicates to deal with the response of the deformable

model with respect to the pre-computed deformation spaces and the interaction requirements.

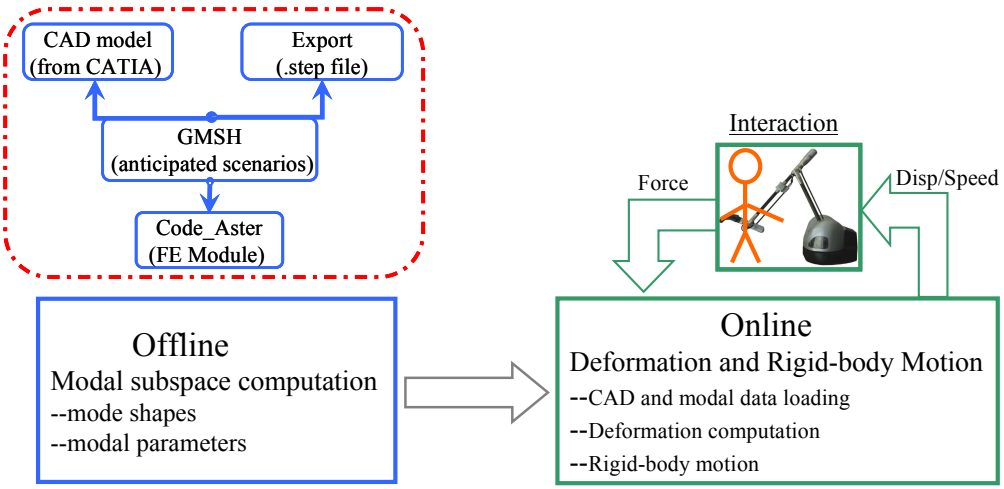


Figure 5.1 – Deformation simulations based on two-stage method: an off-line pre-computation module (left), and a real-time interaction module (right).

Figure 5.1 shows the overall structure of our two-stage deformation simulation method [Wang and Dumont, 2010], which consists of two main phases: an off-line pre-computation phase and a real-time interaction phase. The detailed computation process of both phases is presented in the subsequent sections. As we consider an interaction scenario of integrating the components of rigid-body motion and deformation toward the global deformation verification scenario, the implementation details are explained in section 5.3.3.

5.2 Off-line Pre-computation Phase

Pre-computation is employed extensively in cases of interactive deformation simulations to alleviate the real-time computational load [Barbič, 2007; Peterlík, 2009; Wael et al., 2010a], and different strategies are designed for different pre-computation purposes. In this thesis, the pre-computation phase is dedicated to a data flow aiming at an obtention of deformation spaces in which real-time deformations can be approximated. And in such a way, it is possible to handle the trade-off issue between the deformation accuracy and the real-time haptic interaction performance. As our off-line phase relies on different softwares, the transform of data format is indispensable. One possible improvement over such disadvantage is to integrate the whole pre-computation process in a single interface. The feasible flow of our pre-computation process is illustrated in the left red dash-dot-rectangle in Fig. 5.1. Since the input of the pre-computation is an elastic object, which is defined by the following aspects: (1) a set of tetrahedron elements; (2) several fixed mesh vertices as boundary conditions; and (3) material properties, we present, in the following subsections, our pre-computation phase from three aspects: model preparation, model meshing and deformation space computation.

As we consider the scenario of haptic interaction with deformable and moveable object, rigid-body modes resulting from the pre-computation process are discarded since the rigid-body motion is solved by Equation 4.1 and Equation 4.2 in real-time.

5.2.1 Model Preparation

Our pre-computation process starts on the geometrical modelling in a CAD software (e.g., CATIA™). Considering the format compatibility between the CAD software and the meshing tool, a "step" file is exported from CATIA. It is necessary to point out that the mesh analysis analysis method (see section 4.2.3) and the deformation division scheme (see section 5.3.2) proposed in this thesis are independent of the applied software.

5.2.2 Model Meshing

In the section 2.4.2 of chapter 2, different meshing tools were presented and compared. However, we choose GMSH to mesh our digital models as it integrates the NetGen and TetGen modules. And moreover it is also equipped with pre-processing and post-processing facilities. In this thesis, our mesh model is generated by GMSH, which accepts a file format compatible with "step" format. During this step, the meshing results are stored in two different formats: ".msh" and ".txt". The ".msh" file serve as the input of Code-Aster for the deformation computation process, while the ".txt" file stores the topology of the mesh model. As we do not consider the issues of topology change of deformable objects in this thesis, the ".txt" file stays unchanged during real-time interactions. Due to the inconvenience cased by the length of the mesh file, the content of the file is detailed in the Appendix. Figure A.1 illustrates a meshing result of a beam model and the storage of the topology file, which composes of three components: node (vertex) coordinates, indices of constrain nodes and node connections.

5.2.3 Deformation Space Computation

In our off-line phase, Code-Aster[1] is chosen for modal sub-space computations. Figure A.2 in Appendix shows the Code-Aster interface of a WINDOWS version.

An advantage of choosing Code-Aster as a pre-computation tool is that it provides a free accessibility to the finite element module, and it allows user-defined constraints on the deformable object for different computational purposes. In this section, we test the functionality of the FEM module in Code-Aster with reference to the results from the FEM module in CATIA.

Comparison between CATIA and Code-Aster:

In this subsection, we present a comparison between the FEM module in CATIA and Code-Aster in terms of the modal analysis functionality. The interfaces of both modules are illustrated in section A.1.3 of the chapter Appendix.

As shown in Fig. 5.2, we compare the result of modal frequency resulted from the modal analysis module of Code-Aster and the modal analysis module in CATIA. The two curves almost superimpose with each other and the frequency curve corresponding to the results from CATIA is on the top of the curve of Code-Aster. This can be partially interpreted by the fact that the constraint definition in CATIA is different

[1] www.code-aster.org

from that in Code-Aster. Taking the definition of a fixed surface for example: the whole surface is defined as "fixed" in CATIA, while only finite vertex on the surface is define "fixed" in the command file of Code-Aster. That is to say, the constraint definition in CATIA is more rigid than that in Code-Aster.

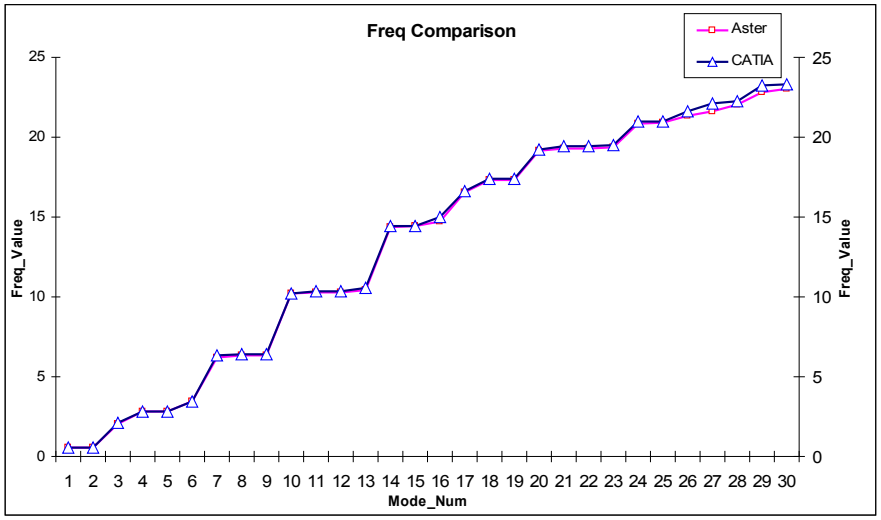


Figure 5.2 – Functionality comparison of modal analysis modules in Code-Aster and CATIA.

Further, the modal frequency error with reference to the results from CATIA is shown in Fig. 5.3. We tested the frequency error concerning 30 modes and the error value is less than 2% for most of the modes. And therefore, the pre-computation results from the FEM module are reliable comparing with the commercial one in CATIA.

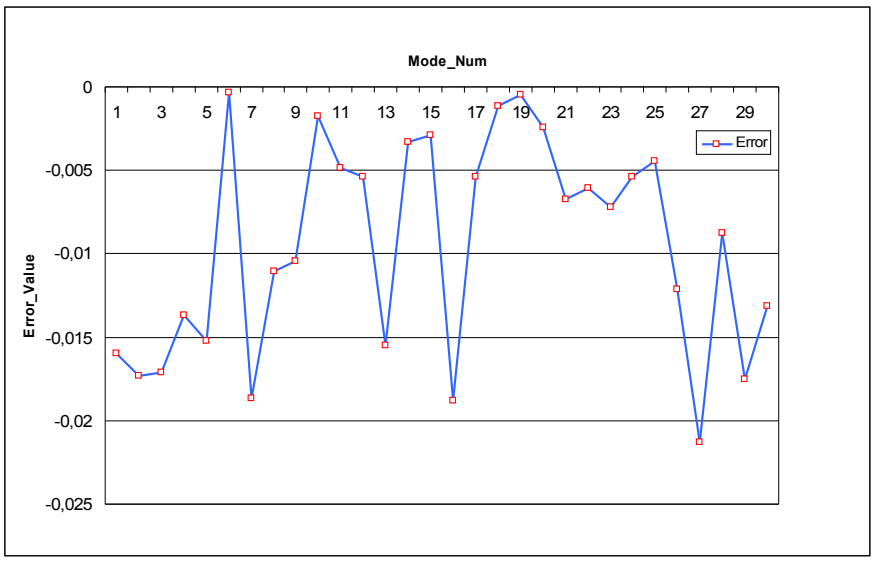


Figure 5.3 – The modal results error of Code-Aster with reference to the results from CATIA.

5.3 On-line Deformation Interaction Phase

After giving the details of the off-line pre-computation phase, we now focus on the implementation of the real-time interaction phase. The main task of this phase is

to compute the real-time responses of the deformable object in terms of two aspects: the realization of the haptic rendering loop and the deformation computation of the whole object. In this thesis, the realization of the haptic rendering loop depends on the admittance control of haptic devices and the real-time interaction steps presented in section 4.3.2. The key blocks of our real-time phase is shown in the right part of Fig. 5.1, starting from the sensation of user's forces, to the computation of modal forces, the computation of reduced displacement and physical displacement and finally to the position and speed feedback for the haptic interfaces.

In the following subsections, we present numerical integration algorithms based on the pre-computed modal deformation space, followed by our on-line deformation division scheme of dividing real-time deformation computation process into two modules which are implemented on two different threads.

5.3.1 Numerical Integration based on Modal Sub-space

There are detailed discussions on the integration of reduced equations in the reference[Hauth et al., 2001]. In this thesis, we choose implicit NewMark integration scheme, as it is a second order accuracy method and the implicit property allows relatively larger time step. These two advantages guarantee better accuracy and better interactive performance. Equation 5.1 and Equation 5.2 express the scheme of implicit NewMark integration employed in the thesis. The decoupled equations are written in an independent algebraic form, as in Equation 5.3.

$$\dot{q}_r^{t+\delta t} = \dot{q}_r^t + [(1-\lambda)\ddot{q}_r^t + \lambda\ddot{q}_r^{t+\delta t}]\delta t \tag{5.1}$$

$$q_r^{t+\delta t} = q_r^t + \dot{q}_r^t\delta t + [(\frac{1}{2}-\mu)\ddot{q}_r^t + \mu\ddot{q}_r^{t+\delta t}]\delta t^2 \tag{5.2}$$

$$m_r\ddot{q_r}^{t+\delta t} + d_r\dot{q_r}^{t+\delta t} + k_r q_r^{t+\delta t} = f_r^{t+\delta t} \tag{5.3}$$

It should be pointed out that $f_r^{t+\delta t} \in f_\Phi$ as expressed in Equation 2.27. One time-step of computing the reduced displacement (q) based on a modal deformation space is presented in **Algorithm 1**. We choose $\lambda = \frac{1}{2}$ and $\mu = \frac{1}{4}$, since these values guarantee the stability of the integration [Basdogan, 2001].

In order to obtain the physical displacement u, a back substitution by Equation 2.20 is mandatory. However, the size of a deformation space Φ grows quickly with the increase of the dimension r or the increase of the complexity of deformable objects. And thus the process of $u(t) = \Phi_r q_r(t)$ becomes the main time consuming task for real-time haptic interactions with modal deformations. In order to obtain a stable and realistic interaction experience, we propose an on-line deformation division scheme which is detailed in the following subsection.

5.3.2 On-line Deformation Division Scheme

In this section, we present the on-line deformation division scheme to meet, on one hand, the hight refresh rate of the haptic rendering loop, and the update rate of the graphical rendering loop on the other. Before the introduction of our proposed division

Algorithm 1 pseudo-code of one time-step of deformation computation based on modal sub-space

Require: CAD data, modal matrix Φ_r, unreduced force vector $\overrightarrow{F}^{t+\delta t}$ at timestep $t+\delta t$, number of modes r, timestep size δt, value q_r, $\dot{q}_r$, $\ddot{q}_r$ at timestep t.
Ensure: value q_r, $\dot{q}_r$, $\ddot{q}_r$ at timestep $t+\delta t$
1: $\overrightarrow{f_r}^{t+\delta t} = [f_1^{t+\delta t}, f_2^{t+\delta t}, ...f_r^{t+\delta t}]^T \leftarrow \Phi_r * \overrightarrow{F}^{t+\delta t}$ {Compute reduced force vector}
2: substitute $\dot{q}_r^{t+\delta t}$ and $q_r^{t+\delta t}$ into Equation 5.3
3: **for** $i = 0$ to r **do**
4: $\alpha_1 = m_i + d_i * \delta t/2 + k_i * \delta t^2/4$, $\alpha_2 = d_i * \delta t/2 + k_i * \delta t^2/4$, $\alpha_3 = d_i + k_i * \delta t$
5: $\ddot{q}_i^{t+\delta t} \leftarrow (f_i^{t+\delta t} - k_i * q_i^t - \alpha_3 * \dot{q}_i^t - \alpha_2 * \ddot{q}_i^t)/\alpha_1$
6: $\dot{q}_i^{t+\delta t} \leftarrow \dot{q}_i^t + (\ddot{q}_i^t * (1-\lambda) + \ddot{q}_i^{t+\delta t} * \lambda) * \delta t$
7: $q_i^{t+\delta t} \leftarrow q_i^t + \dot{q}_i^t * \delta t + ((1/2-\mu) * \ddot{q}_i^t + \mu * \ddot{q}_i^{t+\delta t}) * \delta t^2$
8: **end for**
9: $\overrightarrow{q}^{t+\delta t} \leftarrow [q_1^{t+\delta t}, q_2^{t+\delta t}, ...q_r^{t+\delta t}]^T$
10: $\overrightarrow{u}^{t+\delta t} \leftarrow \Phi_r \overrightarrow{q}^{t+\delta t}$ {Division scheme details in Algorithm 2}
11: return $q_r^{t+\delta t}, \dot{q}_r^{t+\delta t}, \ddot{q}_r^{t+\delta t}$

scheme, we first explain the computational delays issue of coupling heavy deformation computation task with real-time haptic interactions.

5.3.2.1 Computational Delays Issue

The state of the art on deformable object simulations features computational delays. Figure 5.4 illustrates a block diagram of haptic interactions with real-time deformation simulations. The main disadvantage of such interaction scheme is that the update task of HIP position/speed and the graphical rendering task are carried out in the same thread. According to our observation and the discussion in [Barbič and James, 2005], the size of Φ_r grows quickly with the increase of modal subspace dimension r or the increase of model complexity. And thus $U(t) = \Phi_r q(t)$ is the main time consuming process which can not satisfy the high update rate of the haptic rendering loop. In view of such a problem, our on-line deformation division scheme is presented in the following.

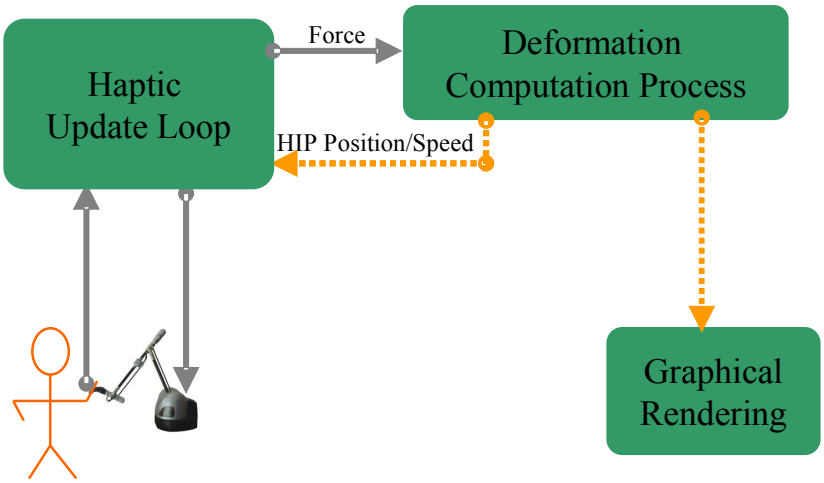


Figure 5.4 – Real-time interaction black diagram without the deformation division scheme.

5.3.2.2 Our Division Scheme

One common practice that allows for haptic rendering at a high refresh rate while interacting with slowly simulated deformable objects is to *decouple* the haptic loop from the graphics and simulation loop. We proposed an on-line deformation division scheme [Wang and Dumont, 2010], as shown in Fig. 5.5. In our division scheme, the deformation computation process is divided into two separate modules (shown by the two green cubes), which are implemented on two different threads. A haptic rendering thread is to update the HIP's position/speed, while a visualization thread is to update the new positions of all vertices. It is necessary to point out that our division scheme is adaptive to the combination of a small deformation and a large rigid-body motion.

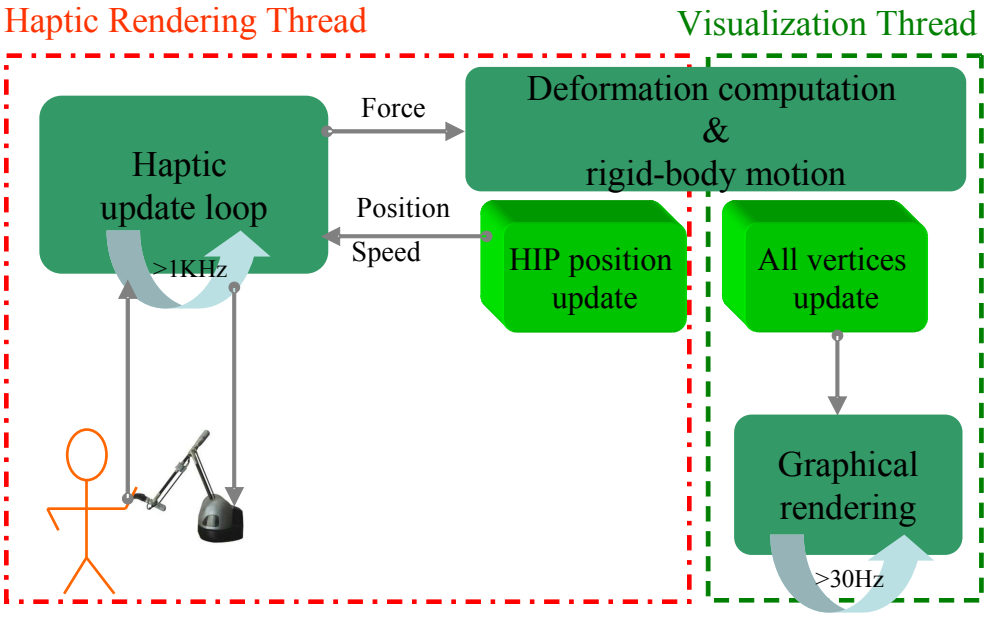


Figure 5.5 – On-line division scheme consists of two separate modules implemented on different threads: HIP position update module to meet high refresh rate of haptic rendering; all vertices update module to meet the lower refresh rate of graphical rendering.

In the *haptic rendering thread*, the following tasks are performed: when HIP "snaps" to an active vertex based on the interaction steps discussed above, the index (e.g. $i^{th}vertex$) of the active vertex is recorded, and the corresponding modal sub-space is chosen according to the real-time switch among modal sub-spaces and then a sub-matrix $\Phi^i = \{\varphi_1^i, \varphi_2^i, ..., \varphi_r^i\}$ is extracted from the pre-computed modal matrix Φ (see Fig. 5.6 showing the extraction); then the forces applied at the end-effector is recorded in $\mathbf{F}$ and the correspnding modal force $\mathbf{f}$ is computed by Equation 2.27; then the response of each mode in terms of the displacement resulting from Equation 2.29 are assembled in $\mathbf{q}$. At last, the displacement and the position of the active vertex are quickly computed by Equation 5.4.

$$\mathbf{u}^i = \Phi^i \mathbf{q}, \quad \mathbf{p}^i = \mathbf{p}_o^i + \mathbf{u}^i \tag{5.4}$$

Here, $\mathbf{p}_o^i$ and $\mathbf{p}^i$ are the undeformed and deformed position of the active vertex, respectively. And the position $\mathbf{p}^i$ is set to update haptic rendering loop at a high refresh rate ($> 1000Hertz$), according to our experiments. The pseudo-code of implementing our division scheme in terms of a sub-matrix abstraction and the feedback computation for

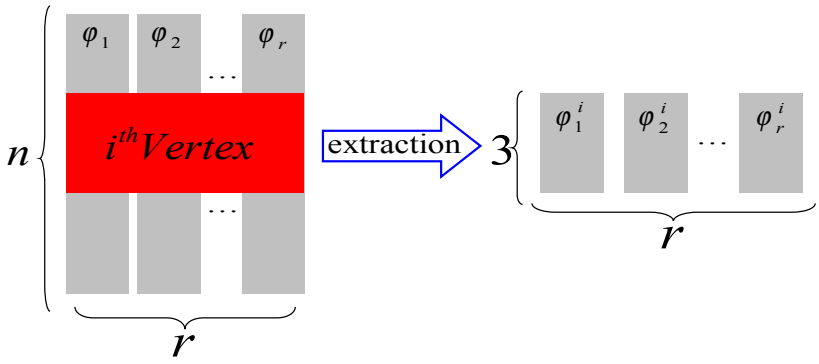


Figure 5.6 – A sub-matrix is extracted from modal matrix considering the index of a single active vertex. For multiple active vertices, e.g. i^{th} and j^{th} vertices, a sub-matrix is extracted like $\Phi^{i,j} = \{\varphi_1^{i,j}, \varphi_2^{i,j}, ..., \varphi_r^{i,j}\}$.

the haptic rendering loop is shown in **Algorithm** 2.

Algorithm 2 Pseudo-code of updating HIP

Require: Interaction surface node index i, number of modes r, reduced displacement $\overrightarrow{q}$ (from the computation in Algorithm 1), HIP initial position $\overrightarrow{P}^i$.
Ensure: HIP deformed position $\overrightarrow{P}^{i'}$

```
for m = 0 to m = 3 do
    for k = 0 to k = r do
        Φ_r^s[k+m*r] ← Φ_r[k+(i+m)*r] {submatrix extraction}
    end for
end for
for j = 0 to j = 3 do
    u_j^i ← Φ_r^s * q_j
    P_j^i' ← P_j^i + u_j^i {update HIP position}
end for
return P⃗^i' ← [P_1^i', P_2^i', P_3^i']^T
```

The *visualization thread* carries out the following two tasks. First, the deformation of the entire object is calculated based on Equation 2.20 and Equation 2.19 by employing the history of reduced displacement vector $\mathbf{q}$, which is passed from the *haptic rendering thread* through a shared memory. Second, the deformations are rendered with a lower refresh rate and the value depends heavily on the meshing densities.

5.3.3 Combination of Deformation and Rigid-body Motion

As we consider the rigid-body motion of a deformable object, our haptic model should be adapted to this case. Figure 5.7 illustrates the key idea of separating and integrating the rigid-body motion and the deformation. Particularly, models of motion and deformation are separately defined based on the two-stage method proposed in this thesis. In our off-line pre-computing phase, the total behaviour that is computed by FEM simulation is separated into components of rigid-body motion and deformation; the component of motion is discarded and the component of deformation which is represented by the modal deformation space is recorded. During our on-line deformation interaction phase, the rigid-body motion is reproduced using the classical equations of

motion, while the deformation is computed based on the two-stage deformation modelling method. And the total behaviour is obtained by combining the components of motion and deformation.

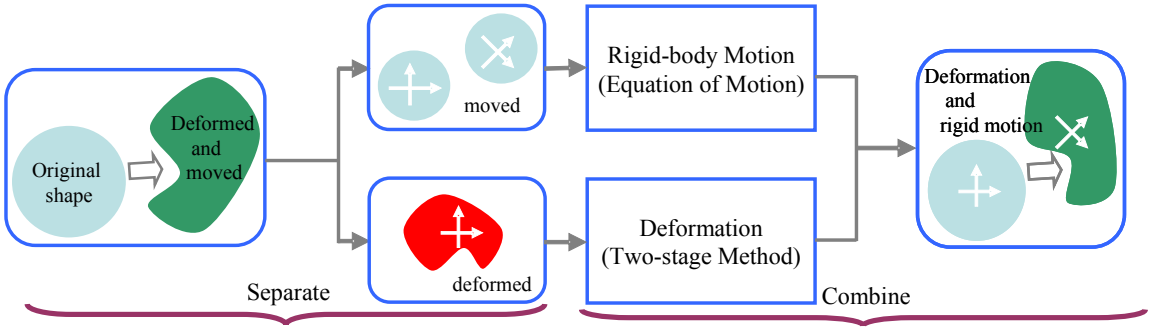


Figure 5.7 – Separation and integration of deformable object's behaviour, including deformations and rigid motions.

Concerning the real-time data flow, Figure 5.8 illustrates the block diagram of the combination of the deformation and the rigid-body motion.

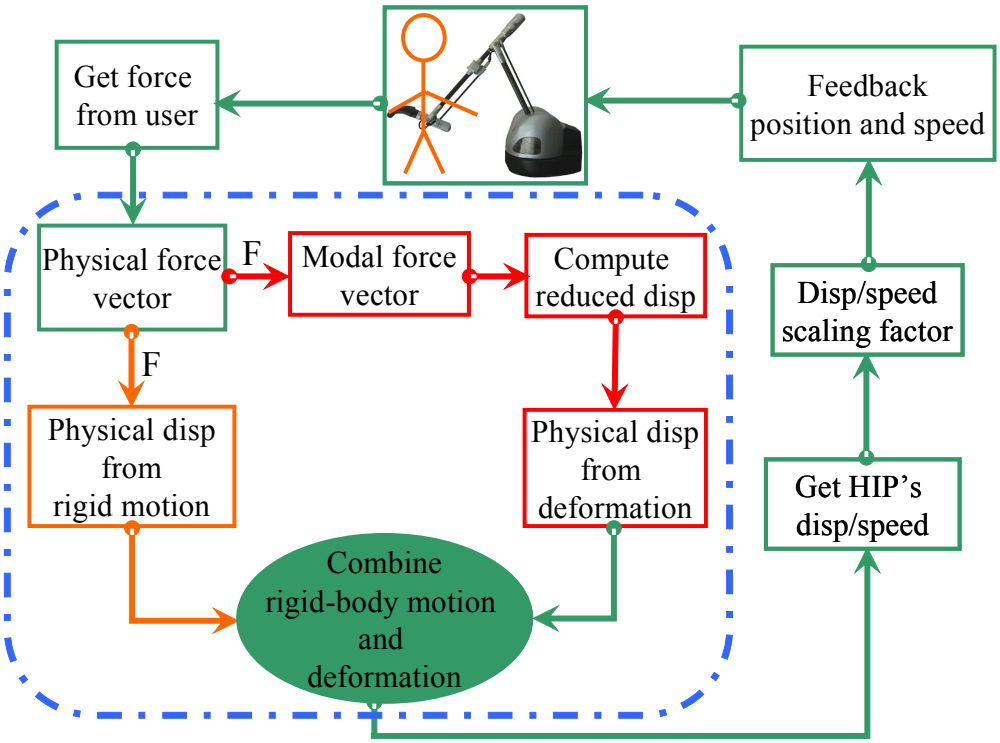


Figure 5.8 – The flow chart of real-time interaction module with deformation and rigid-body motion.

Following the real-time deformation division scheme shown in Fig. 5.5, the "Deformation computation and rigid-body motion" is detailed by the data flow in the frame of blue-coloured dotted-line rectangle in Fig. 5.8. The computational procedure is detailed as follows.

First, the physical forces exerted by the user is recorded using the Virtuose API. As in this thesis we apply single point interaction paradigm, an operator can pull a certain

mesh vertex or a set of vertices in certain directions. Subsequently, the implementation of converting external forces to modal forces can make use of the fact that the user interaction vector is typically sparse.

Second, two computational processes are separated. On one hand, our code prototype orderly calculates modal forces, computes reduced displacement, and obtains physical displacement due to the global deformation. On the other hand, the physical displacement resulting from the rigid-body motion is solved by Equation 4.1 and Equation 4.2.

Third, the rigid-body motion and the global deformation are combined to represent the behaviour of a moveable and deformable object.

5.4 Chapter Summary

In this chapter, we presented our two-stage method in details. Concerning the off-line phase, we described the data flow from the modal preparation, to the meshing procedure, and finally to the computation of modal deformation space. Regarding the on-line phase, we focused on the deformation division scheme which was dedicated to solve the coupling issue of high refresh rate haptic interaction with relatively slow deformation simulation.

In the next chapter, we will present the implementation of our real-time deformation framework as well as experiments concerning the discussion of the trade-off issue. As our goal application is towards the design evaluation of deformable mechanical parts, an industrial feedback is presented.

Implementations and Experiments 6

Contents

In the previous chapters, we explained our deformation simulation framework based on the two-stage method toward the industrial design evaluations of deformable mechanical parts. Our method consists of an off-line pre-computation phase which includes a mesh analysis method, and an on-line deformation interaction phase which includes a deformation division scheme. In this chapter, we present the detailed implementation of the simulation framework. The real-time interaction experiments are presented focusing on the discussion of the trade-off issue between the deformation accuracy and the real-time haptic performance.

6.1 Implementation Configurations

In this section, we introduce the implementation configuration, including the hardware introduction and the descriptions of our experimental models.

6.1.1 Hardware Introduction

Our two-stage method described above and the detailed off-line and on-line techniques have been implemented on the following equipments: Intel® Core™2 Duo CPU 2.6GHz, RAM 3.00GB on Windows XP operating system. Concerning the haptic devices, our implementation is carried out based on two kinds of haptic devices as shown in Fig. 6.1. And the parameter configuration of the two devices are listed in Table 6.1.

(a) Virtuose 6D35-45 (b) Virtuose Desktop

Figure 6.1 – Two kinds of haptc devices employed in our implementation.

Table 6.1 – Parameter configuration of the two haptic devices.

Parameters	device (a)	device (b)
Workspace (mm)	450	120
Maximum Force (N)	35	$7 \sim 10$
Continuous Force (N)	10	$1.4 \sim 3$
Maximum Torque (Nm)	3.1	$0.3 \sim 0.5$
Continuous Torque (Nm)	1	$0.06 \sim 0.14$

6.1.2 Experimental Models

We employed three kinds of models of varying complexity: a simple beam model and a CAD model with three local beams, as shown in Fig. 6.2. In order to clearly explain our off-line mesh analysis method, the bottom three models illustrate three VOIs with denser meshing resolutions. For real-time interactions, the corresponding modal space is switched based on the index of HIP (the red small sphere) and then the model deforms accordingly. The complexity of those models are listed in Table 6.2. And the time required for off-line pre-computations and the data storages are listed at the bottom of the table. The Young's modulus $E = 2200N/m^2$, the Poisson's ratio $\mu = 0.38$, and the density $\rho = 1200kg/m^3$.

6.1.3 Off-line Pre-computation Time and Data Storage

As shown in Table 6.2, we applied three sub-models (i.e., model $(b)(1)$, $(b)(2)$ and $(b)(3)$) for model (b) to test our off-line mesh analysis method and the on-line switch scheme among different deformation spaces corresponding with these sub-models. Although we

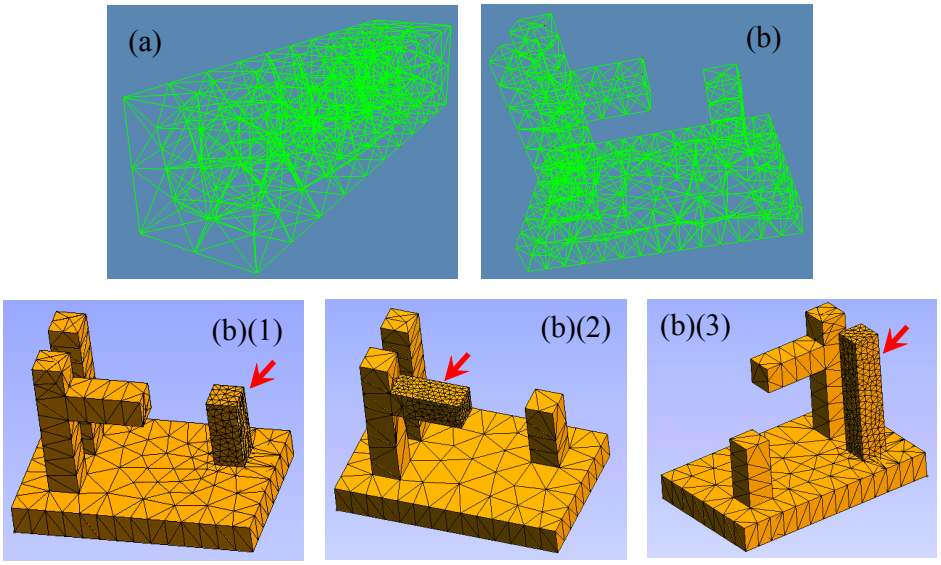


Figure 6.2 – Experimental mesh models of varying complexity: (*a*) simple beam model; (*b*) a CAD coarse mesh model with three local beams.

Table 6.2 – Complexity of our experimental models.

	(a)	(b)	(b)(1)	(b)(2)	(b)(3)
nodes (n)	178	291	630	877	775
tetrahedrons	606	795	2275	3337	2962
off-line (sec)	3.0	8.4	10.0	10.5	9.7
data size (MB)	0.5	0.7	0.8	1.0	0.9

compute deformation spaces in the off-line phase, the deformation modelling process in this thesis is not based on load cases pre-computation, as the ones proposed in [Wael et al., 2010a]. This is because the modal deformation space can describe all the possible small deformations. By contrast, the time needed for our pre-computation is not time-consuming and the data storage is small compared with those in [Tagawa et al., 2009]. This is an advantage of our off-line phase based on the modal deformation spaces pre-computations, which makes it easier to store these modal data and to reproduce rapidly the deformations in the real-time phase.

6.2 Software Development

6.2.1 Overview

Our real-time deformation simulation framework is developed on the platform of Visual Studio 2005 and C++. OpenGL[1] is employed for graphical rendering and Boost C++ library[2] is applied for the multi-thread implementations dedicated to handle the on-line deformation division scheme which is presented in section 5.3.2.

[1] www.opengl.org
[2] www.boost.org

6.2.2 Main Functionalities

The main functionalities of the deformation simulation framework is based on the online block of the two-stage method as illustrated in Fig. 5.1, including loading a mesh model, choosing the dimension of the modal space, coupling the haptic device, computing deformations and implementing the haptic loop. In the following subsections, we present the implementations of these functionalities separately.

6.2.2.1 Loading Mesh Model

A mesh model is loaded first into the deformation simulation framework. The main structure of a mesh file representing a mesh model is presented in section A.1.1.

6.2.2.2 Choosing the Dimension of Modal Space

General principle.
The main benefit of modal deformable model is that the behaviour of the simulation system can be reproduced much more quickly in real-time, as each of the decoupled equations can be solved efficiently. Furthermore, one may check each of the decoupled components and discard those that are irrelevant to the problem at hand. For example, removing modes that are too stiff or too high frequency to be observed will not change the appearance the resulting simulation, while removing them will greatly reduce the computational cost.

Mode choice discussion concerning the visual and haptic perception.
It is crucial to indicate that the general principle of choosing deformation modes should be adaptive to haptic interaction applications, because with the increase of the mode frequency, their actual contributions to the deformation visualization is smaller. Visible deformations resulting from lower frequency modes, e.g., from a number of 20 modes to 30 modes, are necessary for visualizations, according to our interaction experience. However, the story is slight different when the haptic interaction with deformable modal model is taken into account. The reason is based on the fact that the influence of higher frequency modes to the real-time haptic interaction applications. On one hand, real-time projection-based model reduction for deformable objects has, however, suffered from an important limitation: the reduction basis, e.g., modal deformation basis in our case, is global in space and time. Such bases require a large number of modal vectors to capture local deformations [Barbič and Zhao, 2011]. On the other hand, with the increase of the modal basis dimension, more high frequency mode content tend to enter the solution, and explicit integration time-step is progressively limited. This is because a necessary condition for the explicit integrator to be stable is that the time-step be stable to represent the oscillations of the highest eigenfrequency of the linearised reduced system–the higher frequency, the smaller the time-step has to be. And it is also the reason why implicit integration scheme is generally applied [Wriggers, 2003; Barbič and James, 2005].

High frequency modes tend to damp more quickly, and often perceptionally important for only a short time. Afterwards, simulating only lower frequency modes can suffice to capture perceptually important dynamics. The different visual effects of the

choice of mode is shown in Fig. 6.6, while the haptic perceptions of mode-dependent deformation simulations are not explored in this thesis. This direction of research would be an important part of future work. A source of reference could be the work presented in [Zheng and James, 2011], in which the authors extensively presented the high quality modal sound synthesis by investigating mode-adaptive simulations and by defining adaptive criterions of increasing or decreasing the number of mode.

6.2.2.3 Coupling Haptic Device

The connection between a haptic device and a deformation simulator present the data flow between the two components as we described in section 4.1.4 concerning our haptic deformation model. In this thesis, we apply admittance control for the haptic coupling as shown in Fig. 6.3.

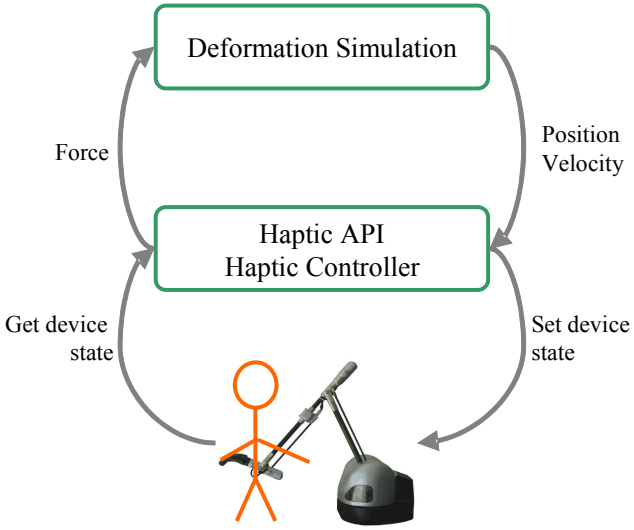


Figure 6.3 – Overview of a haptic interface control in admittance mode.

To maintain the consistency of haptic coupling with the deformation simulation, the haptic device needs to be controlled properly. As we discussed in chapter 4, the admittance control of a haptic interface proposes to apply the following scheme: the simulation is responsible for setting the state of the device (the position, orientation, linear and angular velocities), while the device returns forces to be integrated into the simulation for the next step.

6.2.2.4 Computing Deformations and Graphical Rendering

According to the algorithm 1, the reduced displacement $\mathbf{q}$ is computed first and then the corresponding physical displacement is computed by the back substitution through a multiplication of the modal matrix and the reduced displacement vector. Then the deformation of the object can be obtained with respect to the rest position of each mesh vertex, and then the graphical rendering is followed.

6.2.2.5 Implementing Haptic Loop and Haptic Rendering

Following the deformation division scheme introduced in section 5.3.2 and the computational task in the *haptic rendering thread*, the position and velocity of the HIP is

computed based on which the hapic state is set.

6.3 Interaction Scenario I: Fixed Deformable Objects

After the explanations of the main functionalities of our deformation simulation framework, we focus on the real-time interaction scenarios concerning the experimental models presented in section 6.1.2. Our interaction scenarios are based on the single-point interaction model and the interaction steps discussed in section 4.3.2.

In this section, we discuss two scenarios concerning the real-time haptic interactions: haptic interactions with deformable objects fixed in the spatial space for the purpose of local deformation verification scenarios, and haptic interactions with deformable and moveable object for the purpose of global deformation verifications scenarios.

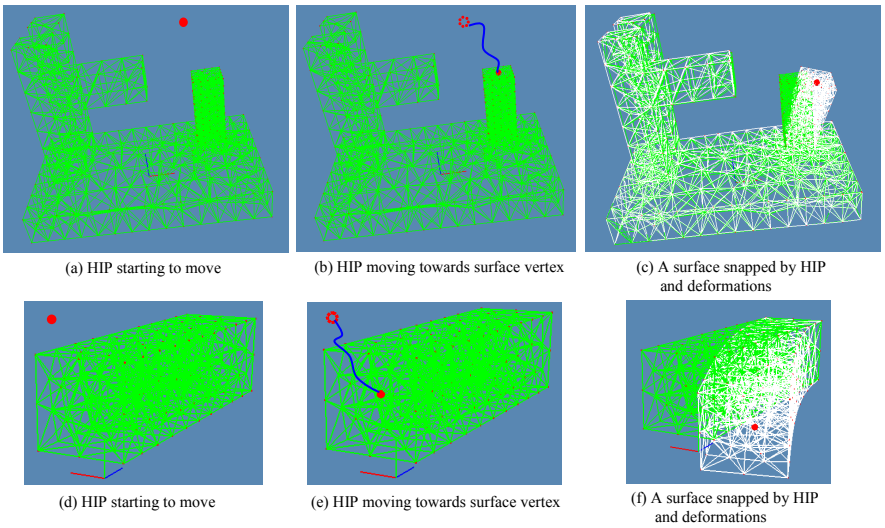
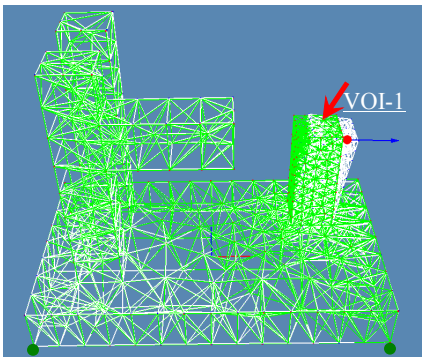


Figure 6.4 – Real-time interaction scenario of model (*b*) (top) and the beam model (bottom).

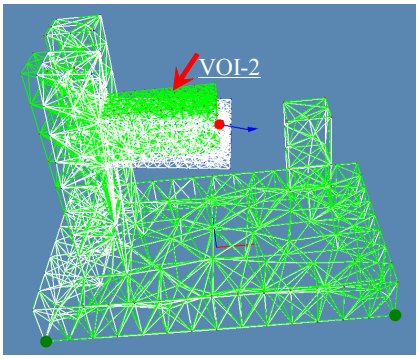
As illustrated in Fig. 6.4, the real-time interaction scenarios concerning the model (*b*) and the beam model are described in three steps. First, the HIP is connecting with the end-effector of the haptic device based on the instruction in subsection 6.2.2.3. Second, the HIP is moving towards a surface vertex following the nature movement of a user's hand (see the blue curves). And finally a surface vertex is "snapped" by the HIP and the on-line deformation interaction phase starts (see section 5.3).

6.3.1 Switch Scheme of the Mesh Analysis Method

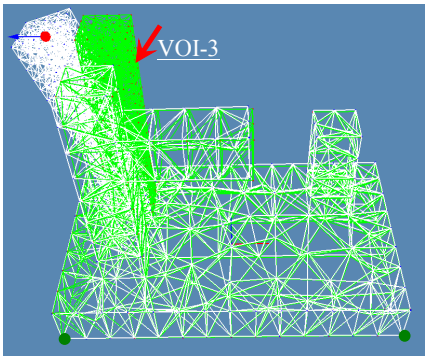
Our mesh analysis method is applied on the model (*b*) concerning three VOIs with denser meshes, as shown in Fig. 6.5. For real-time interactions, the corresponding modal deformation space is switched based on the index of the HIP (the red small sphere) and then the model deforms accordingly. Particularly, the picture (a) shows that the modal deformation space corresponding to the VOI-1 is chosen to represent the real-time deformation interactions when the HIP "snaps" to a vertex in the region VOI-1 pointed by an arrow in red color. The similar interactions happen concerning the scenarios illustrates in the picture (b) and the picture (c).



(a) Interaction with VOI-1



(b) Interaction with VOI-2



(c) Interaction with VOI-3

Figure 6.5 – Our mesh analysis method is applied on model (b) concerning three VOIs with denser meshes. The green meshes are at rest position, while the white ones are deformed meshes.

It is important to point out that our on-line deformation space switch scheme is based on the pre-computations of the different global deformation spaces concerning the VOIs as shown in Fig. 4.4. Our switch scheme is different from the real-time progressive and adaptive meshing technique in that we mesh a deformable model based on the anticipated interaction scenarios, resulting in different mesh representations of the same deformable model. And the modal deformation spaces are globally pre-computed based on those VOIs. The essence of the real-time switch on the deformation spaces is based on the location of the HIP. If the HIP "snaps" with a mesh vertex located in one of the VOIs, the corresponding deformation space is used to describe real-time deformations.

6.3.2 Deformation Accuracy Discussion

As we discussed in the previous sections, the trade-off between the deformation accuracy and the interaction performance is an essential issue for haptic interactions with deformable mechanical parts towards the design evaluation process in the early industrial design phase. In this subsection, we discuss the choice of choosing modal vectors and their effects on the deformation accuracy.

As illustrated in Fig. 6.6, we have chosen $2, 5, 10, 20$ and 30 modes to represent different modal deformation spaces concerning model (a) – the beam model. The displacement of a vertex in the Y direction is tested and the comparison with the corresponding result from *non-reduced deformation space* is accomplished. The red

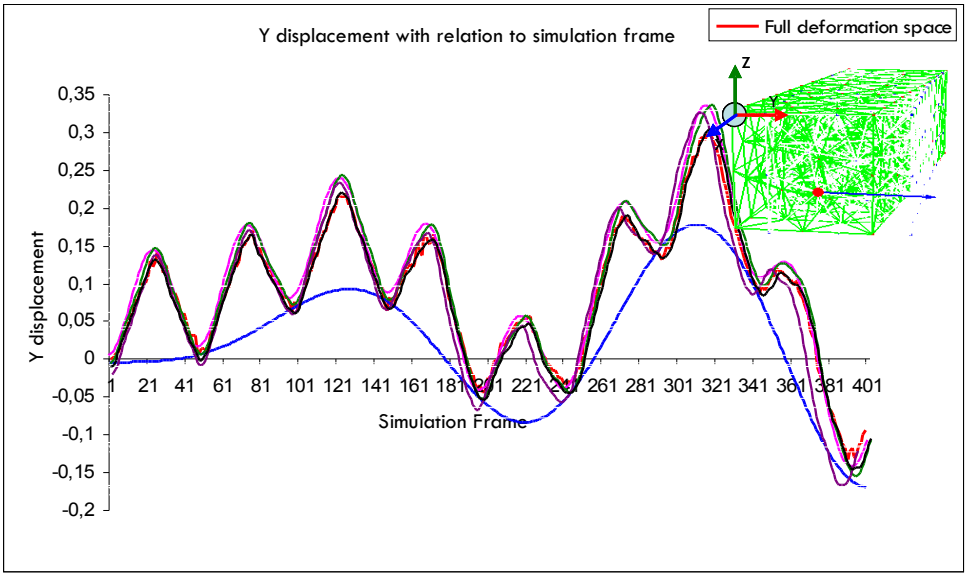


Figure 6.6 – Displacement comparison of one vertex on model (a) resulting from different modal sub-spaces, including non-reduced deformation space.

curve describes the vertex displacement based on a non-reduced deformation space, while the blue one corresponding to the results based on modal sub-space composed of 2 modes obviously deviates. However, the other curves are almost superimposed. According to our real-time interaction experience, with the increasing number of modes (e.g., from 20 modes to 30 modes) entering into the deformation sub-space, more detailed deformations, such as twistings, are observed.

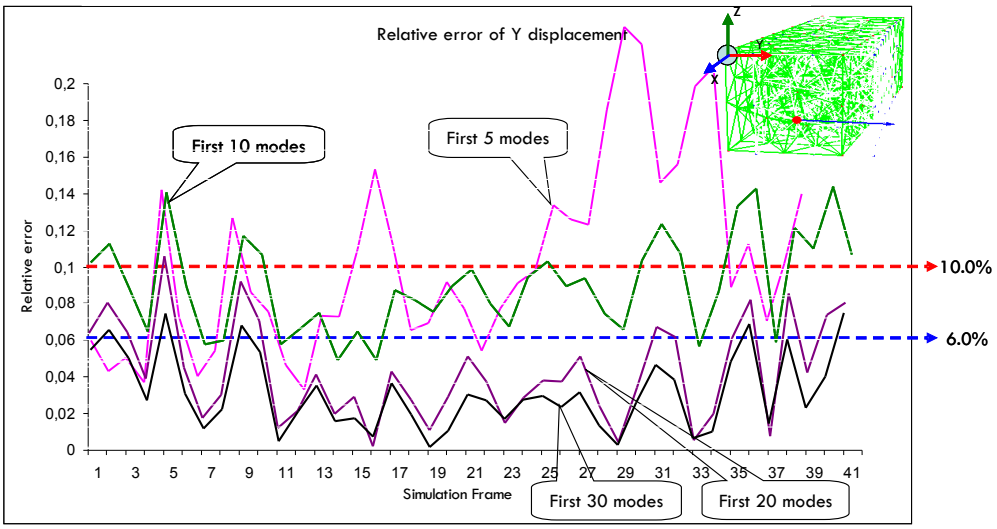


Figure 6.7 – Relative error of X displacement of one node based on different modal subspaces.

In order to further illustrating the difference of those superimposed curves shown in Fig. 6.6, we defined a relative error as the difference percentage with reference to the displacement results from a full deformation space. The curves in Fig. 6.7 illustrate the values of relative errors with relation to different modal subspaces. The relative error of the displacement resulting from the modal subspace of first 5 modes is large than 10.0%, And the relative error of the displacement resulting from the modal subspace

of first 10 modes is less than 10.0%, while larger than 6.0%. By contrast, regarding the modal subspace of first 20 modes and first 30 modes, the relative error of the displacement is less than 6.0%. From this comparison, a scalable modal deformation accuracy can be achieved by choosing different number of modes or different modes.

6.3.3 On-line Interaction Performance Results

In this section, we discuss the update rates of the haptic rendering thread and the visualization thread regarding these different experimental models.

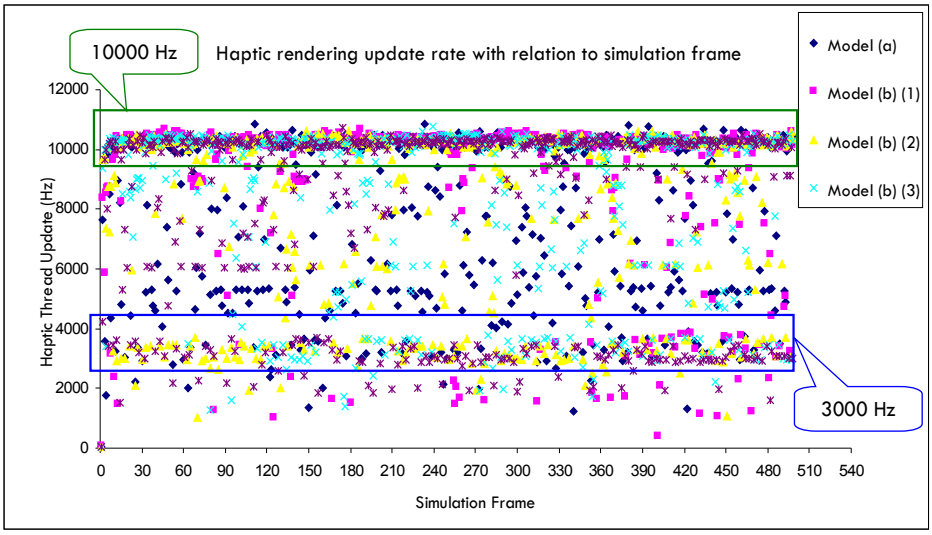


Figure 6.8 – Haptic thread update rate concerning different models varying complexities.

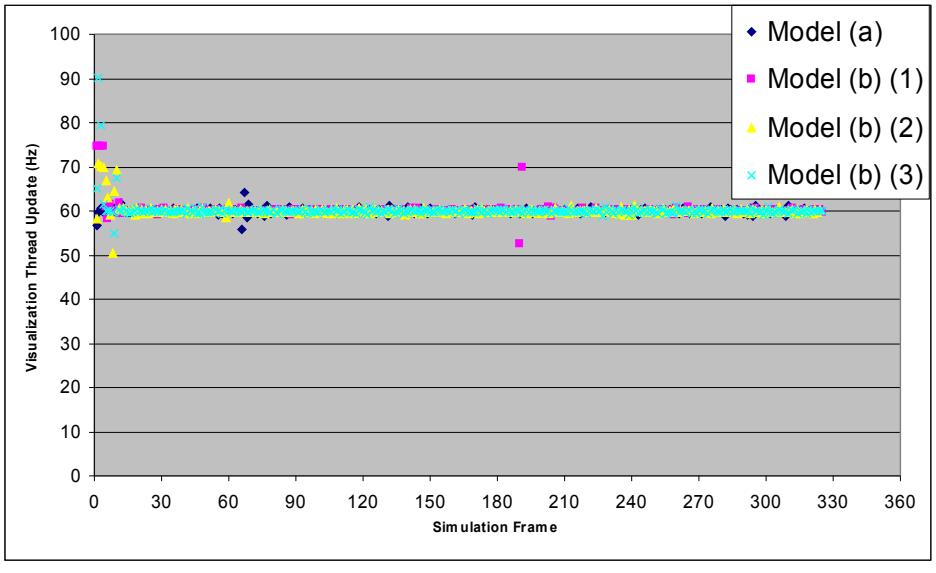


Figure 6.9 – The update rate of our visualization thread which includes the computation of all vertices' new position and the rendering process on graphic card.

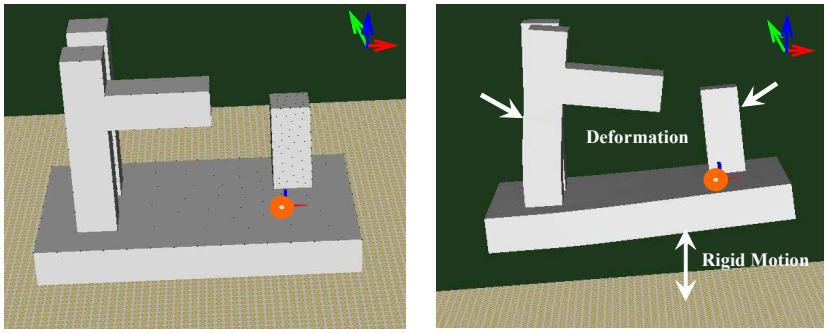
On one hand, as illustrated in Fig. 6.8, due to our on-line division scheme which implements HIP updates and all vertices updates on different threads, the haptic thread

update rate is around 10K Hertz, which allows smooth interactions with our experimental models. We observe that this value is almost independent on the model's complexity, even for the complex model (c). This is a significant feature of our real-time deformation model based on the division scheme, as we extract a sub-matrix (as shown in Fig. 5.6) to compute the position and speed of the active vertex, and moreover, the size of the sub-matrix is $3 * r * sizeof(float)$ given a single point interaction. Therefore, our haptic loop can be quickly updated with a stable interaction experience.

On the other hand, the update rate of the visualization thread is shown in Fig. 6.9. For the model (a), the model b and its sub-models, we obtain around a value of 60 Hertz, which is sufficient for interaction applications. It should be noted that the update value of the visualization thread depends heavily on the complexity of an experimental model and the meshing resolution of the model.

6.4 Interaction Scenario II: Moveable and Deformable Objects

Figure 6.10 illustrates a manipulation scenario with a moveable and deformable model: (a) shows the model's original position on the floor, and (b) shows the model's final position which results from a combination of a small deformation and a rigid-body motion. The HIP in this case is represented by the sphere in orange color.



(a) Rest pose (b) Deformation and rigid motion

Figure 6.10 – Haptic manipulation scenario with a moveable and deformable object: (a) rest pose of the object; (b) the combination of a small deformation and a large rigid motion.

Figure 6.11 displays the interaction force when a user lifts a deformable object from a floor. As the movement of the model is starting from the floor, the interaction force curve (blue one) in Z direction can be divided into three zones. The zone 1 is the "raising phase" where the interaction force is increasing as the model is lifted slowly from the floor. The zone 2 is the "keeping phase" where the interaction force fluctuates without considerable changes. In this phase, the deformable object is free-floating. The zone 3 is the "releasing phase" where the interaction force returns to zero when the haptic device disconnects. The other two curves in green and red color illustrating interaction forces in X and Y directions have low force amplitude, since the main rigid-body motion of the model is in Z direction.

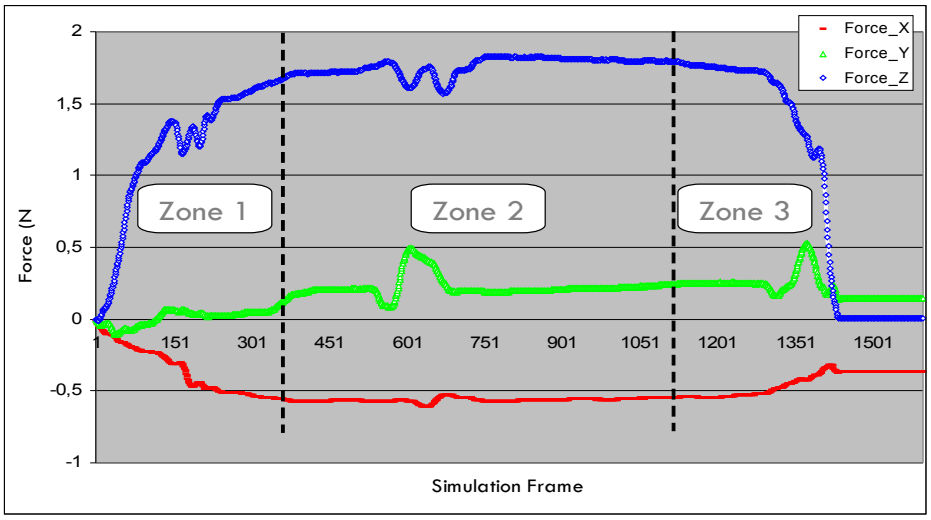


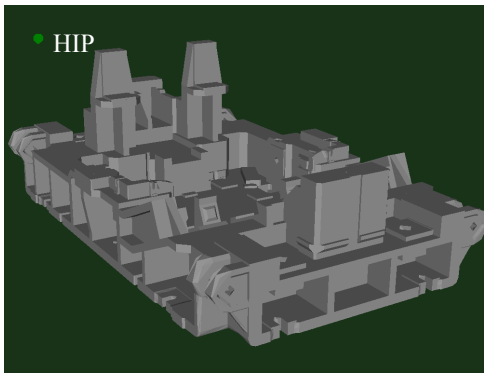
Figure 6.11 – Interaction force when a user lifts a moveable and deformable object from a floor, the same scenario as shown in Fig. 6.10.

6.5 Preliminary Results on Industrial CAD Model

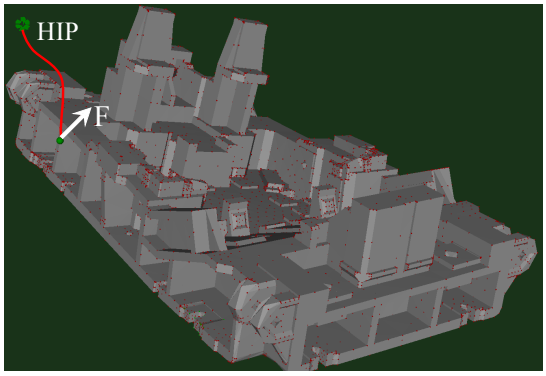
In this section, we present some preliminary results concerning the real-time haptic interaction with an industrial CAD model, which is the same model as shown in Fig. 1.9 and Fig. 1.10 for the purpose of illustrating our industrial verification scenarios. We follow the same procedures of the off-line pre-computation phase and the on-line interaction phase to represent a deformation model for the real-time haptic manipulation purpose. This model is meshed by 21499 mesh vertices and 88198 tetrahedrons and the data storage of the pre-computation data is 137M bytes.

Figure 6.12 shows the real-time interaction scenarios of the industrial model. The picture (a) illustrates the CAD model at a rest pose and the green sphere as a HIP to represent the end-effector of the haptic interface. The picture (b) shows a global deformation scenario where the HIP snaps to a surface mesh vertex by following the natural movement of user's hand (see the curve in red color), while the picture (c) presents a local deformation scenario and the arrow "F" indicates the force direction applied by a user through the haptic interface.

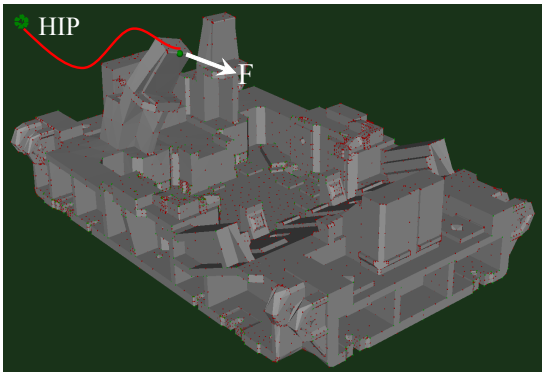
The experimental results of this industrial model illustrates that the update rate of the haptic rendering loop keeps high (more than 1000 Hertz) thanks to the on-line deformation division scheme. However, due to the complexity of the CAD model, the update rate of the visualization thread decreases to 10 Hertz which poses problem for interactive applications. As we will discuss in the future work, this problem could be resolved either by simplifying the geometrical model or by extending our mesh analysis method to a multi-fold representation of complex deformable objects. And the exploration of GPU's computational and rendering power is equally a promising solution [Yinghui et al., 2006].



(a) Industrial CAD model



(b) Global deformation



(c) Local deformation

Figure 6.12 – Real-time haptic interaction scenario of an industrial CAD model.

6.6 Experiment Bench

An experimental bench is developed based on the details of off-line pre-computations and on-line deformation interactions discussed in the previous sections.

Figure 6.13 illustrates real-time haptic interactions with an experimental deformable object through the haptic device Virtuose™6D35-45. Within the framework of the industrial project EMOA, these interaction experiments are carried out in the virtual reality room of PSA Peugeot Citroën who is the host of the project.

Four serial interactions with a deformable part are illustrated in Fig. 6.13: (a) the configuration of connecting the haptic device and the deformation simulator; while $(b), (c)$ and (d) indicate the pull, press and push operations respectively. According to the interaction experience, the user can feel the resistance of the deformable object while deforming it through the hand's natural movement on the end-effector of the haptic device.

6.7 Chapter Summary

In this chapter, we described the implementation of our real-time deformation simulation framework by introducing the hardware configuration, the statistical data of our experimental models and the software development in terms of the main functionalities. Then we discussed the experimental results from two aspects regarding the

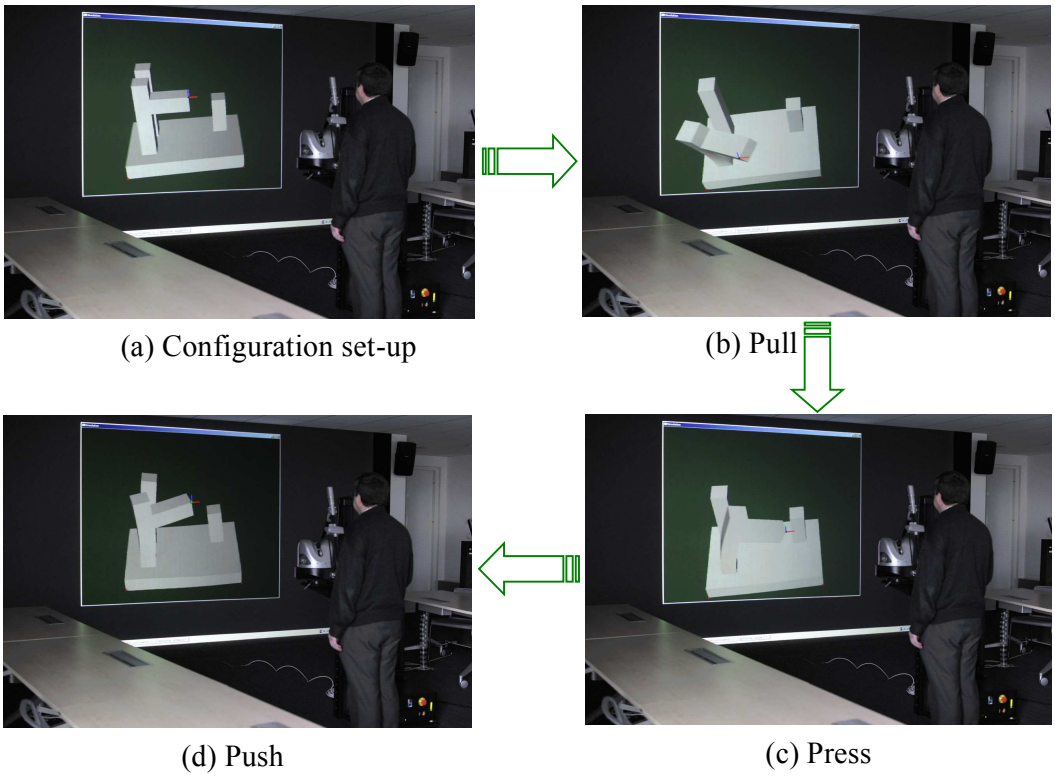


Figure 6.13 – Our experimental bench: different interaction operations with a deformable mechanical part through a haptic interface (courtesy of PSA Peugeot Citroën).

different models and the different number of modes: the deformation accuracy and the real-time interaction performance. Concerning the real-time interactions, we presented a scenario of haptic interaction with fixed deformable objects for the purpose of illustrating the local deformation verification scenarios, while the other scenario of haptic interaction with moveable and deformable objects showed the corresponding global deformation verification scenarios. At last, we presented our experimental bench with the cooperation of PSA.

In the next chapter, a conclusion of the thesis work is made and several perspectives of future work are probed.

Conclusion

The objective of our work was to investigate the real-time deformation simulation through haptic interfaces for the purpose of the design evaluation of deformable mechanical parts towards the process of an industrial PLM.

We have first presented a review of related work on the integration of the VR technology and the industrial PLM. This review focused on three main aspects. We first studied the components of the PLM and pointed out the position of our research work in such a background. Then, we exposed the problems of current CAE tools for the process of the PLM and argued that the employment of the VR technology for the PLM could contribute to the efficient data flow in terms of design information among different industrial sectors. Finally, we overviewed the VR–PLM integration applications aiming to identify the problematic issues of a product design considering the infeasibility of assemblying two pars, the defect of the general appearance, and the flaw of ergonomics.

This state of the art underlined the necessity of investigating the process of the design validation of deformable mechanical parts through haptic interfaces in a VR environment.

Then, we have presented the theoretical background of the deformation modelling methods and the mathematical formulation of the finite element method. And we focused on the descriptions of the model reduction method and the modal analysis method. We continued to overview the foundations of haptics in a VR environment by emphasizing the importance of employing haptic interfaces for VR–PLM integrations while the key challenges of haptic interactions with deformable mechanical parts in terms of the trade-off issue between the deformation accuracy and the real-time interaction performance. Then, we presented a survey of relevant work on real-time deformation modelling methods which were dedicated to handle the trade-off issue. We stressed that the model reduction method was more attractive for the real-time haptic interaction with deformable mechanical parts which were formulated by the FEM. Moreover, we explained the advantages of our two-stage method over the general deformation modelling methods based on the model reduction.

Based on the aforementioned state-of-the-art, we focused on our work on the the development of a real-time deformation simulation framework for haptic interactions.

First, we explained our haptic deformation model by presenting a detailed computational model describing the data flow between the deformation modelling on one hand and the haptic interfaces on the other. On one hand, we presented the main components of finite element modelling followed by the introduction of our off-line

mesh analysis method based on anticipated evaluation scenarios and the on-line switch scheme among different modal deformation spaces. On the other hand, the single point interaction model and our real-time interaction steps were described in detail.

Second, we presented the implementation of our real-time deformation simulation framework by introducing the hardware configuration, the statistical data of our experimental models and the main functionalities of the software development. Experimental results showed that our two-stage method could efficiently handle the trade-off issue, as the deformation modelling was formulated by the finite element method which guarantees the deformation accuracy. And moreover, the heavy computations of large elastic systems were occurred off-line which assured a costless deformation response model and thus a stable real-time haptic interaction performances were achieved.

Future work

In this thesis, we proposed a two-stage method for the small deformation modelling and combine it with the large rigid-body motion for the purpose of design evaluation of industrial deformable part, aiming at dealing with the trade-off between the deformation accuracy and the interaction performance within an industrial context. In our method, the off-line pre-computation phase provides a costless deformation response model for the real-time interactions. Moreover, our mesh analysis method based on anticipated scenarios allows the real-time deformation to be computed with an accuracy enhancement on degrees of freedom where they are necessary. And our on-line division scheme ensures the haptic thread to be refreshed at more than 1000 Hertz which provides a smooth interaction experience according to our experiments.

There are several potential industrial applications that could benefit from the deformation modelling and the real-time interaction method proposed in this paper. One example is the real-time deformation simulation for the integration of interactive analysis and re-design in a VR environment [Fischer et al., 2009]. Another example is to introduce volumetric deformations into the procedure of virtual assemblies or maintenance operations guided by haptic interfaces.

However, the future work of improving or extending our method will be mainly focused on the following three aspects.

First, concerning the low interactive rate of complex industrial CAD models, one possible solution is to simplify the geometry before the off-line pre-computations, since the geometry simplification is a common pre-process of finite element analysis [Foucault et al., 2004]. Another possible solution is to extend our mesh analysis method to a multi-fold representation of deformable objects [Seiler et al., 2010].

Second, we limit the interaction possibilities to a selection of possible manipulations, which means that the pre-computation process must be re-carried out for a new interaction scenario, as our two-stage method depends heavily on the prescribed boundary conditions. It is an important future work to extend the capacity of our method to allow arbitrary interactions by considering the time-variant boundary conditions [Hauser et al., 2003] or by generating the "pre-computation" date during the interaction by distributed computational resources [Filipovič et al., 2009].

Third, the visualization of the deformation is not sufficient for mechanical indus-

tries, and thus some other results are equally required, e.g., stress and strain fields. This is truly important for interactive deformation analysis applications, because the conversion from the stress state to a value for a haptic device could be an enhancement to aid designers better understand how a part deforms in the early design phase.

Appendix

A

Contents

THIS appendix provides more details of our pre-computation phase in terms of the content of command files for Code-Aster and modal result files.

A.1 Pre-computation Procedure in Code-Aster

As presented in section 5.2.3 of chapter 5, due to the inconvenience caused by the length of the command file and the modal results file, we describe, in the following subsections, the pre-computation procedure in Code-Aster. For the purpose of clarity, we take a beam model to describe the procedures of pre-computing deformation spaces.

A.1.1 Mesh File

The first step is to prepare a mesh model. Figure A.1 illustrates the mesh model of the beam.

The mesh topology file is shown in the following file, which keeps unchanged during real-time haptic interactions.

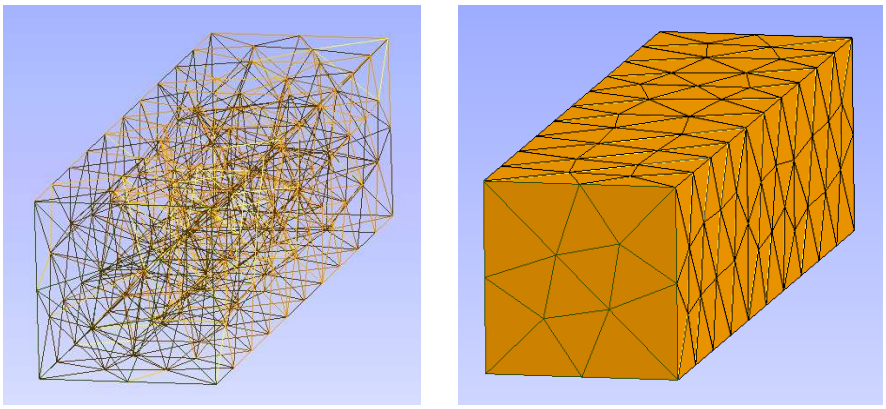


Figure A.1 – A mesh model of a beam. Left: Wire-frame model. Right: Mesh model with volume faces.

Deformable Object Topology File

```
$NOD
178
1 -5.34957695007 29.6701087952 0      ← node coordinate
2 -5.34957695007 29.6701087952 100
3 -5.34957695007 4.67010879517 0
4 -5.34957695007 4.67010879517 100
5 19.6504230499 4.67010879517 100
6 19.6504230499 4.67010879517 0
7 19.6504230499 29.6701087952 0
8 19.6504230499 29.6701087952 100
 ......
 ......
 ......
174 14.43506163244483 23.19240820852117 37.93040158111326
175 1.053949455021631 21.96715517149279 24.4921629878957
176 2.108118075993935 11.1620683776262 13.15904774576722
177 13.20793734178255 11.15509340746931 49.85052231333333
178 0.9455120849249468 22.37407297608768 13.34117005000644
$ENDNOD

$ELM
606
3
1 2 1 6 3 29 139 5
2 2 1 6 3 50 5 139
3 2 1 6 3 140 8 50
15 4 2 1 4 67 71 88 146   ← a tetrahedron element
16 4 2 1 4 137 59 70 144
17 4 2 1 4 66 127 58 146
18 4 2 1 4 88 71 141 146
19 4 2 1 4 76 67 88 146
......
......
......
598 4 2 1 4 82 60 26 169
599 4 2 1 4 46 95 47 129
600 4 2 1 4 116 124 44 174
601 4 2 1 4 55 65 56 164
602 4 2 1 4 135 137 39 161
603 4 2 1 4 144 137 161 176
604 4 2 1 4 144 59 70 176
605 4 2 1 4 161 137 70 176
606 4 2 1 4 137 144 70 176
$ENDELM
```

A.1.2 Command File

The following file shows a serial of Code-Aster commands to compute modal parameters and modal vectors, which form a modal matrix Φ as described in section 2.3.3.

Commands of Code_Aster for Computing Deformation Space

```
DEBUT(IMPR_MACRO = 'OUI',);

PRE_GMSH();                                                    Loading Mesh
MAIL = LIRE_MAILLAGE();

IMPR_RESU(UNITE = 99,
          RESU = _F(MAILLAGE = MAIL,),) ;

MAIL = DEFI_GROUP(reuse  = MAIL,
                  MAILLAGE = MAIL,
                  CREA_GROUP_MA = (_F(NOM = 'BODY',
                                      GROUP_MA = 'GM1',),),
                  CREA_GROUP_NO = ( _F(NOM = 'FIXED1',                  Constrained
                                       GROUP_MA = 'GM1',),              Vertex
                                    _F(NOM = 'FIXED2',
                                       GROUP_MA = 'GM2',),
                                    _F(NOM = 'FIXED3',
                                       GROUP_MA = 'GM3',),),),);

MATER = DEFI_MATERIAU(ELAS = _F(E = 2.2E03,                    Material Property
                                NU = 0.38,
                                RHO = 1200E-12,),);

MODM = AFFE_MODELE(MAILLAGE = MAIL,
                   AFFE = _F(TOUT = 'OUI',
                             PHENOMENE = 'MECANIQUE',
                             MODELISATION = '3D',),);

CHMAT = AFFE_MATERIAU(MAILLAGE = MAIL,
                      AFFE = _F(TOUT = 'OUI',
                                MATER = MATER,),);

CH1 = AFFE_CHAR_MECA(MODELE = MODM,
                     DDL_IMPO = (_F(GROUP_NO = 'FIXED1',
                                    DX = 0.0,
                                    DY = 0.0,
                                    DZ = 0.0, ),

                                 _F(GROUP_NO = 'FIXED2',
                                    DX = 0.0,
                                    DY = 0.0,
                                    DZ = 0.0,),

                                 _F(GROUP_NO = 'FIXED3',
                                    DX = 0.0,
                                    DY = 0.0,
                                    DZ = 0.0,),),);
```

- 1 -

```
Me = CALC_MATR_ELEM(OPTION = 'MASS_MECA',
                    MODELE = MODM,
                    CHAM_MATER = CHMAT,
                    CHARGE = CH1,);

Ke = CALC_MATR_ELEM(OPTION = 'RIGI_MECA',
                    MODELE = MODM,
                    CHARGE = CH1,
                    CHAM_MATER = CHMAT,);

vect = CALC_VECT_ELEM(CHARGE = CH1,
                      CHAM_MATER = CHMAT,
                      OPTION = 'CHAR_MECA',);

NU = NUME_DDL(MATR_RIGI = Ke);

M = ASSE_MATRICE(MATR_ELEM = Me,NUME_DDL = NU);
K = ASSE_MATRICE(MATR_ELEM = Ke,NUME_DDL = NU);
VECT = ASSE_VECTEUR(VECT_ELEM = vect,NUME_DDL = NU);

DEFI_FICHIER(ACTION    = 'ASSOCIER',
             FICHIER   = 'C:/ASTER/16Dec/Mass_Matrix',
             UNITE     = 66,);

DEFI_FICHIER(ACTION    = 'ASSOCIER',
             FICHIER   = 'C:/ASTER/16Dec/stiffness_Matrix',
             UNITE     = 50,);

DEFI_FICHIER(ACTION    = 'ASSOCIER',
             FICHIER   = 'C:/ASTER/16Dec/mode_results',
             UNITE     = 16,);

DEFI_FICHIER(ACTION    = 'ASSOCIER',
             FICHIER   = 'C:/ASTER/16Dec/2',
             UNITE     = 51,);

Mode = MODE_ITER_SIMULT(MATR_A = K,
                        MATR_B = M,
                        METHODE = 'TRI_DIAG',
                        OPTION = 'MODE_RIGIDE',);

Resul = EXTR_MODE(
                  FILTRE_MODE = _F(MODE = Mode,
                  NUME_MODE = (r),),);

IMPR_RESU(UNITE = 16,
          RESU  = ( _F(RESULTAT   = Resul,
                       TOUT_ORDRE = 'OUI',
```

Elementary Matrix

Assembly Matrix

Define Output Files

Modal Analysis & Extracting ***r*** mode

```
                                TOUT_PARA  = 'OUI',
                                TOUT_CHAM  = 'OUI',)),
                                FORMAT = 'RESULTAT',);

DEFI_FICHIER(ACTION = 'LIBERER',
                        UNITE = 16,);

STANLEY();

FIN();
```

Close Files
&
End Program

A.1.3 Modal Analysis Interface

Here we present two interfaces for implementing modal analysis: the module in Code-Aster and the module in CATIA. We choose Code-Aster due to its open-source feature and free accessibility to the definition of the modal analysis process. The comparison of both modules are explained in section 5.2.3.

Code-Aster Interface:

Once the preparation of the model is finished, the modal pre-computation occurs based on the introduction of the code-aster interface. Figure A.2 shows the interface of code-aster for windows.

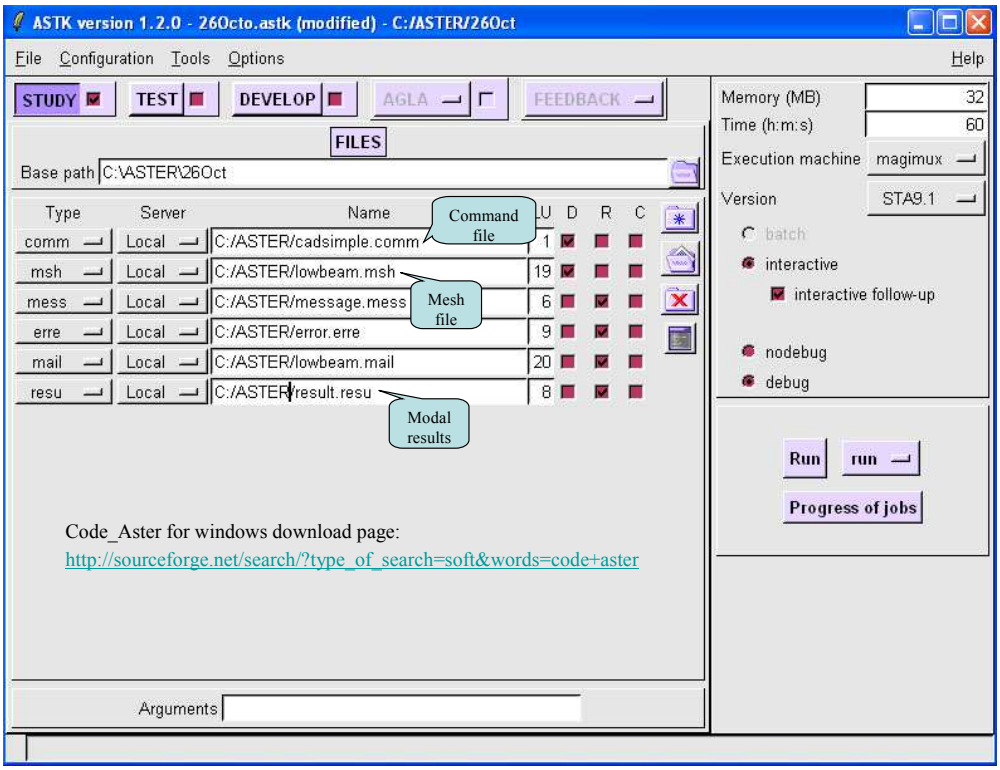


Figure A.2 – Code-Aster interface for WINDOWS.

As shown in Figure A.2 are two inputs file are required for the Code-Aster: a mesh file and a command file. The preparation of a mesh file can follow the discussion in the previous subsections, while the command file is a script file composed of a sequential of commands defined by the Code-Aster. For the computation of linear deformation space, these commands include the following functions, e.g., loading a mesh file, defining material properties (i.e. density, Young's modulus, Poisson's ratio, etc.), specifying boundary conditions, implementing linear modal analysis, etc. More detailed description of the commands can be found in the Appendix. The results from Code-Aster include modal parameters of each mode and the corresponding mode shape vector, and these mode vectors form the modal matrix Φ (see Figure 4.2) which is employed to described the real-time deformation responses.

CATIA Interface:

The commercial CAE software CATIA also provides a module for carrying out the modal analysis. Figure A.3 illustrates the interface of implementing the modal analysis on a simple beam example. The modal results from both interfaces are compared and shown in Figure 5.2.

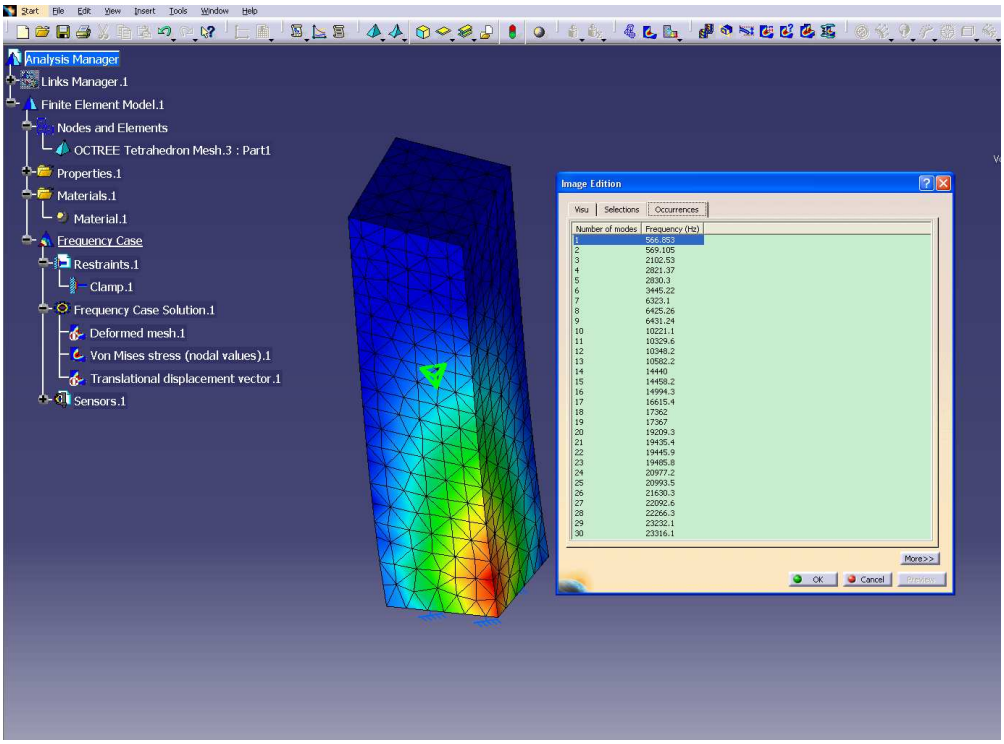


Figure A.3 – The interface of modal analysis module in CATIA.

A.1.4 Modal Results File

With respect to the instructions of performing pre-computations in section 5.2, the modal results file is shown in the followings. It should be noted that $\Phi = \{\varphi_1, \varphi_2, ..., \varphi_r\}$, while $\varphi_1 = \{\varphi_1^1, \varphi_1^2, ..., \varphi_1^n\}$. Here, r is the number of mode chosen to describe real-time deformation responses and n is the number of mesh vertex. In this example case, $n = 630$.

Modal Results from Code_Aster

IMPRESSION DES PARAMETRES DU CONCEPT Resul
POUR LE NUMERO D'ORDRE 1

NUME_MODE	***1***
FREQ	6.65824E+02
OMEGA2	1.75016E+07
AMOR_REDUIT	0.00000E+00
MASS_GENE	1.16227E-05
RIGI_GENE	2.03416E+02
AMOR_GENE	0.00000E+00
MASS_EFFE_DX	7.28357E-06
MASS_EFFE_DY	3.53874E-06
MASS_EFFE_DZ	2.24634E-05
FACT_PARTICI_DX	7.91624E-01
FACT_PARTICI_DY	5.51787E-01
FACT_PARTICI_DZ	-1.39022E+00
MODELE	MODM
CHAMPMAT	CHMAT
CARAELEM	
TYPE_DEFO	
NOEUD_CMP	
EXCIT	
NORME	SANS_CMP: LAGR

Modal Parameters of Mode ***1***

------>
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE DEPL
NUMERO D'ORDRE: 1 NUME_MODE: 1 FREQ: 6.65824E+02

NOEUD	DX	DY	DZ	
N1	7.61238E-01	3.94594E-01	-2.49191E-01	← φ_1^1
N2	7.65591E-01	4.72099E-01	-8.09376E-01	
N3	7.96576E-01	4.72317E-01	-7.11586E-01	
N4	7.88642E-01	3.94777E-01	-1.52590E-01	
N5	5.70537E-01	2.95716E-01	-1.49048E-01	
N6	5.71644E-01	3.74409E-01	-7.11585E-01	
N7	5.40731E-01	3.74375E-01	-8.09485E-01	
N8	5.44741E-01	2.97581E-01	-2.44002E-01	
N9	7.59399E-02	4.08410E-02	-7.19835E-02	
N10	9.71981E-01	4.65089E-01	-4.03513E-02	
N11	7.42662E-02	4.10662E-02	-3.23517E-02	
N12	9.99214E-01	4.65587E-01	5.67604E-02	
N13	1.00381E-01	3.67819E-02	-3.02156E-02	
N14	6.61103E-02	3.21194E-02	2.93577E-02	
N15	2.47036E-02	1.27187E-01	-3.80377E-02	← φ_1^{15}
N16	3.70487E-02	1.19209E-01	-6.51099E-01	

N17	6.69766E-02	4.89880E-02	-1.08220E-01
N18	6.96490E-02	5.06513E-02	-1.59106E-01
N19	4.10835E-02	1.16665E-01	-4.11351E-01
N20	4.10947E-02	1.13972E-01	-5.20284E-01
N21	4.42393E-02	1.06891E-01	-3.88101E-01
N22	4.44352E-02	1.06259E-01	-4.89572E-01
……			
……			
……			
N612	9.75881E-02	3.22303E-01	-4.48184E-01
N613	1.01248E-01	3.30800E-01	-4.10156E-01
N614	8.95778E-02	2.97250E-01	-4.87872E-01
N615	7.62259E-02	2.39687E-01	-4.47330E-01
N616	8.13925E-02	2.54257E-01	-4.09361E-01
N617	3.44581E-02	8.45142E-02	-3.76744E-01
N618	6.64334E-02	2.04746E-01	-4.61818E-01
N619	3.20712E-02	5.86282E-02	-1.33030E-01
N620	3.55027E-02	2.56067E-02	-2.09613E-01
N621	4.82931E-02	1.34888E-01	-4.49140E-01
N622	3.96720E-02	4.79220E-02	-1.33058E-01
N623	8.51453E-02	2.74730E-01	-4.50141E-01
N624	3.55924E-02	7.10156E-02	-3.98790E-01
N625	6.12416E-01	3.98457E-01	-7.57713E-01
N626	3.60552E-02	8.76896E-02	-4.82009E-01
N627	3.05566E-02	6.65681E-02	-1.70377E-01
N628	6.27812E-02	1.93600E-01	-4.85177E-01
N629	4.25172E-02	2.38036E-02	-3.21298E-02
N630	3.20795E-02	7.06717E-02	-4.89321E-01

φ_1^{630}

======>

IMPRESSION DES PARAMETRES DU CONCEPT Resul
POUR LE NUMERO D'ORDRE 2

NUME_MODE	***2***
FREQ	8.45470E+02
OMEGA2	2.82200E+07
AMOR_REDUIT	0.00000E+00
MASS_GENE	7.46735E-06
RIGI_GENE	2.10728E+02
AMOR_GENE	0.00000E+00
MASS_EFFE_DX	2.16411E-06
MASS_EFFE_DY	6.90837E-06
MASS_EFFE_DZ	3.07507E-07

Modal Parameters of Mode ***2***

- 2 -

```
          FACT_PARTICI_DX     -5.38340E-01
          FACT_PARTICI_DY     9.61844E-01
          FACT_PARTICI_DZ -2.02929E-01
          MODELE          MODM
          CHAMPMAT        CHMAT
          CARAELEM
          TYPE_DEFO
          NOEUD_CMP
          EXCIT
          NORME           SANS_CMP: LAGR

------>
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE  DEPL
NUMERO D'ORDRE: 2 NUME_MODE: 2 FREQ:   8.45470E+02
NOEUD       DX          DY          DZ
N1      -7.32009E-01  7.82717E-01  3.92486E-02   <-- φ2^1
N2      -7.46447E-01  1.00000E+00  6.56572E-01
N3      -6.59415E-01  9.99840E-01  8.61170E-01
N4      -6.56614E-01  7.82831E-01  2.42244E-01
N5      -4.20569E-01  5.77062E-01  2.33876E-01
N6      -4.10752E-01  7.95010E-01  8.61413E-01
N7      -4.97178E-01  7.94721E-01  6.56446E-01
N8      -4.93434E-01  5.72247E-01  3.50069E-02
N9      -2.06862E-02  4.32014E-02 -2.84121E-02
N10     -9.56403E-01  9.14320E-01 -1.84770E-01
N11     -2.52021E-02  4.78402E-02  3.38479E-02
N12     -8.84791E-01  9.13344E-01  1.65877E-02
N13      2.07410E-02  2.81241E-02 -3.15207E-02
N14     -3.20589E-02  4.26122E-02  2.73415E-02
N15      1.83845E-02  5.29038E-02  5.37299E-03   <-- φ2^15
N16      1.22762E-02  4.80627E-02 -2.81986E-01
N17     -1.46738E-02  5.02234E-02  8.05123E-02
N18     -9.56415E-03  4.38162E-02  6.78165E-03
N19      1.19577E-02  4.79443E-02 -1.37670E-01
......
......
......
N624     1.60075E-03  2.71100E-02 -1.10989E-01
N625    -5.16471E-01  8.45112E-01  7.55901E-01
N626     5.47734E-03  3.44267E-02 -1.66752E-01
N627    -3.55890E-04  2.14187E-02 -4.11316E-02
N628     3.54839E-02  9.26879E-02 -1.69885E-01
N629    -1.23472E-02  2.48079E-02  9.79654E-02
N630     1.68611E-03  2.65568E-02 -1.63250E-01   <-- φ2^630
```

A.1.5 Unit in Code-Aster

As we discussed in section A.1.2, the material properties of the deformable object were defined in the command file. The units of some physical quantities are listed here.

A Correspondance dans les unités

Unités	METRIQUE MKS	METRIQUE mmNS	ANGLO-SAXONNE FPS	ANGLO-SAXONNE IPS
Longueur	m	mm	ft	in
Temps	sec	sec	sec	sec
Masse	Kg	tonne	slug	lbf-sec^2
Force	N	N	lbf	lbf
Température	°C	°C	°F	°F
Aire	m^2	mm^2	ft^2	in^2
Volume	m^3	mm^3	ft^3 (cu-ft)	in^3 (cu-in)
Vitesse	m/sec	mm/sec	ft/sec	in/sec
Accélération	m/sec^2	mm/sec^2	ft/sec^2	in/sec^2
Angle, rotation	rad	rad	rad	rad
Vitesse angulaire	rad/sec^2	rad/sec^2	rad/sec^2	rad/sec^2
Masse volumique	Kg/m^3	Tonne/mm^3	slug/ft^3	lbf-sec^2/in^4
Moment, couple	N-m	N-mm	ft-lbf	in-lbf
Force linéïque	N/m	N/mm	lbf/ft	lbf/in
Force répartie sur une surface (Contrainte, pression,Module d'Young)	N/m^2 (Pa)	N/mm^2 (MPa)	lbf/ft^2	lbf/in^2 (Psi)
Coefficient de Dilatation thermique	/°C (/K)	/°C (/K)	/°F (/K)	/°F (/K)
Moment Quadratique d'une poutre IG_z	m^4	mm^4	ft^4	in^4
Moment d'inertie transverse d'une poutre	Kg-m^2	tonne-mm^2	slug-ft^2	lbf-in-sec^2
Energie, Travail, Chaleur	J	mJ	ft-lbf	in-lbf
Puissance, taux de transfert thermique	W	mW	ft-lbf/sec	in-lbf/sec
Gradient de température	°C/m	°C/mm	°F/ft	°F/in
Flux thermique	W/m^2	mW/mm^2	lbf/ft-sec	lbf/in-sec
Conductivité thermique	W/m-°C	mW/mm-°C	lbf/sec-°F	lbf/sec-°F
Chaleur spécifique C$_p$	J/Kg-°C	mJ/tonne-°C	ft-lbf/slug-°F	in^2/sec^2-°F

Rappels :

- 1 W ≡ 1 N-m/sec,
- 1 mJ ≡ 1 N-mm,
- 1 mW ≡ 1 N-mm/sec,
- 1 N/m^2 ≡ 1 Pa,
- 1 N/mm^2 ≡ 1 MPa,

A.2 Screen-shots of Real-time Single-point Haptic Interactions

Figure A.4 – Six screen-shots of single-point haptic interaction with a CAD model of three local beams. The active node is depicted by green bullet with a three coordinate lines.

A.3 Real-time haptic interaction videos

I have put several real-time haptic interaction videos in my website followed the link bellow: https://sites.google.com/site/zhaoguangwang2011/home. And these videos can also be found on the Youtube.

Author's Publications

[1] Zhaoguang WANG, Georges DUMONT. Deformation Computation of CAD Models for Haptic Manipulation. *In Association Française de Réalité Virtuelle (AFRV'09)*, Lyon, France, December 9-11, 2009.

[2] Zhaoguang WANG, Georges DUMONT. Real-time Deformation Verification of Industrial CAD Model in VR with Haptic Interaction. *Poster Presentation in Virtual Reality International Conference (VRIC'10)*, Laval, France, April 2010.

[3] Zhaoguang WANG, Georges DUMONT. Real Time Interaction with Deformable Industrial CAD Model through Haptic Interface in VR. *In Proceedings of IDMME-Virtual Concept 2010*, Bordeaux, France, 2010.

[4] Zhaoguang WANG, Georges DUMONT. Interactive Design Validation of Deformable Parts Through Haptic Interface. *In Proceedings of the ASME World Conference on Innovative Virtual Reality (WINVR2011)*, Milan, Italy, June 27-29, 2011.

[5] Zhaoguang WANG, Georges DUMONT. Haptic Manipulation of Deformable CAD Parts with a Two-stage Method. International Journal on Interactive Design and Manufacturing (IJIDeM). (under review)

Bibliography

Adams, R. and Hannaford, B. (1999). Stable haptic interaction with virtual environments. *Robotics and Automation*, 15(3):465–474. 69

André, B. and Delingette, H. (2008). Versatile design of changing mesh topologies for surgery simulation. In *Proceedings of the 4th international symposium on Biomedical Simulation*, pages 147–156. 65

Aziz, F. A. and Mousavi, M. (2009). A review of haptic feedback in virtual reality for manufacturing industry. *Journal of Mechanical Engineering*, 40(1):68–71. 16

Bai, Z. and Li, R.-C. (2005). Structure-preserving model reduction using a Krylov subspace projection formulation. *Comm. Math. Sci.*, 3(2):179–199. 55

Balaniuk, R. and Salisbury, K. (2002). Dynamic simulation of deformable objects using the long elements method. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 58–65. 28

Bangerth, W., Hartmann, R., and Kanschat, G. (2007). deal.II - a general purpose object oriented finite element library. *ACM Trans. Math. Softw*, 33(4). 42

Barbagli, F., Salisbury, K., and Prattichizzo, D. (2003). Dynamic local models for stable multi-contact haptic interaction with deformable objects. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 109–116. 21

Barbič, J. (2007). *Real-time Reduced Large-Deformation Models and Distributed Contact for Computer Graphics and Haptics*. PhD thesis, Computer Science Department School of Computer Science Carnegie Mellon University. 72

Barbič, J. and James, D. L. (2005). Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):982–990. vi, 37, 38, 56, 58, 76, 84

Barbič, J. and Zhao, Y. (2011). Real-time large-deformation substructuring. *ACM Trans. on Graphics (SIGGRAPH 2011)*, 30(4):911–917. 84

Basdogan, C. (2001). Real-time simulation of dynamically deformable finite element models using modal analysis and spectral lanczos decomposition methods. In *Proceedings of the Medicine Meets Virtual Reality*. 75

Basdogan, C., De, S., Kim, J., Muniyandi, M., Kim, H., and Srinivasan, M. (2004). Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64. 28

Basdogan, C., Laycock, S. D., Day, A. M., Patoglu, V., and Gillespie, R. B. (2007). *3-DOF haptic rendering*. A. K. Peters. 50

Basdogan, C. and Srinivasan, M. A. (2001). *Haptic Rendering In Virtual Environments*. Lawrence Erlbaum Associates. 50

Bazen, A. M. and Gerez, S. H. (2003). Fingerprint matching by thin-plate spline modelling of elastic deformations. *Pattern Recognition*, 36(8):1859–1867. 27

Berkelman, P. J., Butler, Z. J., and Hollis, R. L. (1996). Design of a hemispherical magnetic levitation haptic interface device. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 17–22. 48

Bianchi, G., Harders, M., and Székely, G. (2003). Mesh topology identification for mass-spring models. In *MICCAI 2003*, volume 1, pages 50–58. 53

Bianchi, G., Solenthaler, B., Székely, G., and Harders, M. (2004). Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2004*, volume 2, pages 293–301. Springer. 53

Bookstein, F. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567–585. 27

Boothroyd, G. and Dewhurst, P. (1989). *Product design for assembly*. McGraw-Hill, New York. 13

Bordegoni, M. and Cugini, U. (2006). Haptic modeling in the conceptual phases of product design. *Virtual Reality*, 9(2-3):191–202. 16

Bordegoni, M., Cugini, U., Belluco, P., and Aliverti, M. (2009). Evaluation of a haptic-based interaction system for virtual manual assembly. In *Proceedings of the 3rd International Conference on Virtual and Mixed Reality*, pages 303–312, Berlin, Heidelberg. Springer-Verlag. 15

Böttcher, G., Allerkamp, D., and Wolter, F.-E. (2010). Multi-rate coupling of physical simulations for haptic interaction with deformable objects. *The Visual Computer: International Journal of Computer Graphics*, 26:903–914. 54

Bourdot, P., Convard, T., Picon, F., Ammi, M., Touraine, D., and Vézien, J. (2010). VR-CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. *Computer-Aided Design*, 42(5):445–461. 17

Brough, J. E., Schwartz, M., Gupta, S. K., Anand, D. K., Kavetsky, R., and Pettersen, R. (2007). Towards the development of a virtual environment-based training system for mechanical assembly operations. *Virtual Reality*, 11:189–206. 15

Bullinger, H., Breining, R., and Bauer, W. (1999). Virtual prototyping - state of the art in product design. In *26th International Conference on Computers and Industrial Engineering*, pages 103–107. 6

Burdea, G. (2003). *Virtual Reality Technology*. 2 edition. 11

Burdea, G. C. (1996). *Force and touch feedback for virtual reality*. New York: Wiley. 18, 46

Choi, M. G. and Ko, H.-S. (2005). Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 11:91–101. 58

Colgate, J. E., Stanley, M. C., and Brown, J. M. (1995). Issues in the haptic display of tool use. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3. 70

Comas, O., Taylor, Z. A., Allard, J., Ourselin, S., Cotin, S., and Passenger, J. (2008). Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework sofa. In *ISBMS*, pages 28–39. 55

Convard, T. (2005). *Conception assistée par ordinateur en environnement immersif*. PhD thesis, Université Paris Sud XI. 15

Coquillart, S. (1990). Extended free-form deformation: a sculpturing tool for 3D geometric modeling. *SIGGRAPH Computer Graphics*, 24:187–196. 27

Corso, J. J., Chhugani, J., and Okamura, A. M. (2002). Interactive haptic rendering of deformable surfaces based on the medial axis transform. In *Proceedings of Eurohaptics*, pages 92–98. 53

Cotin, S., Delingette, H., and Ayache, N. (1999). Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5:62–73. 33, 51

Cover, S., Ezquerra, N., O'Brien, J., Rowe, R., Gadacz, T., and Palm, E. (1993). Interactively deformable models for surgery simulation. *IEEE Computer Graphics Applications*, 13:68–75. 27

Cretu, A.-M. and Petriu, E. (2006). Neural-network-based adaptive sampling of three-dimensional-object surface elastic properties. *Instrumentation and Measurement, IEEE Transactions*, 55(2):483–492. 57

Dai, F., Felger, W., Frühauf, T., Göbel, M., Reiners, D., and Zachmann, G. (1996). Virtual prototyping examples for automotive industries. In *Virtual Reality World*. 9, 12

Daunay, B., Micaelli, A., and Regnier, S. (2007). 6-DOF haptic feedback for molecular docking using wave variables. In *IEEE International Conference on Robotics and Automation*, pages 840–845. 51

Daunay, B. and Regnier, S. (2009). Stable six degrees of freedom haptic feedback for flexible ligand-protein docking. *Computer-Aided Design*, 41(12):886–895. 51

de Pascale, M., de Pascale, G., Prattichizzo, D., and Barbagli, F. (2004). A GPU-friendly method for haptic and graphic rendering of deformable objects. In *Proceedings of Eurohaptic2004*. 55

de Pascale, M., Mulatto, S., and Prattichizzo, D. (2008). Bringing haptics to Second Life. In *Proceedings of the 2008 Ambi-Sys workshop on Haptic user interfaces in ambient media systems*. 51

Debunne, G., Desbrun, M., Barr, A. H., and Cani, M.-P. (1999). Interactive multiresolution animation of deformable models. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*. vi, 53, 54

Debunne, G., Desbrun, M., Cani, M.-P., and Barr, A. H. (2001). Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36. ACM. 67

Deitz, D. (1995). Real engineering in a virtual world. *Mechanical engineering*, 117:78–85. 12

Delingette, H., Cotin, S., and Ayache, N. (1999). A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Proceedings of Computer Animation*, pages 70–81. 65

Deussen, O., Kobbelt, L., and Tücke, P. (1995). Using simulated annealing to obtain good nodal approximations of deformable bodies. In *Sixth Eurographics Workshop on Simulation and Animation*, pages 30–43. 53

Dulong, J., Druesne, F., and Villon, P. (2007). A model reduction approach for real time part deformation with non-linear mechanical behaviour. *International Journal on Interactive Design and Manufacturing*, 1:229–238. 59

Dupont, T., Hoffman, J., Johnson, C., Kirby, R. C., Larson, M. G., Logg, A., and Scott, L. R. (2003). The FEniCS project. Technical Report 21, Chalmers Finite Element Center Preprint Series. 42

Eddy, J. and Lewis, K. E. (2002). Visualization of multidimensional design and optimization data using cloud visualization. In *Proceeding of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2002)*, pages 30–43. 16

Elhelw, M., Chung, A., Darzi, A., and Yang, G.-Z. (2003). Image-based modelling of soft tissue deformation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 2878, pages 83–90. Springer Berlin/Heidelberg. 27

Etzmuss, O., Gross, J., and Strasser, W. (2003). Deriving a particle system from continuum mechanics for the animation of deformable objects. *IEEE Transactions on Visualization and Computer Graphics*, 9:538–550. 53

Faas, D., Fischer, A., and M.Vance, J. (2007). Interactive mesh-free stress analysis for mechanical design assembly with haptics. In *ASME 2007 International Design Engineering Technical Conferences, DETC07-34660*, Las Vegas. 50

Filipovič, J., Peterlík, I., and Matyska, L. (2009). On-line precomputation algorithm for real-time haptic interaction with non-linear deformable bodies. In *Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 24–29. 96

Fischer, A., Vance, J. M., and Vo, D. M. (2009). Haptic feedback to guide interactive product design. In *World Conference on Innovative Virtual Reality (WINVR09)*. 15, 17, 21, 46, 96

Fong, P. (2009). Sensing, acquisition, and interactive playback of data-based models for elastic deformable objects. *The international Journal of Robotics Research*, 28(5):630–655. 57

Fong, T., Thorpe, C., and Baur, C. (2000). Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and web-based tools. In *Vehicle Teleoperation Interfaces Workshop, IEEE International Conference on Robotics and Automation*. 11

Foucault, G., Marin, P.-M., and Léon, J.-C. (2004). Mechanical criteria for the preparation of finite element models. In *13th International Meshing Roundtable*, pages 413–426. 96

Galoppo, N., Otaduy, M. A., Mecklenburg, P., Tekin, S., Gross, M., and Lin, M. C. (2007). Haptic rendering of high-resolution deformable objects. In *proceedings of 12th International Conference on Human-Computer Interaction*. 54

Garat, A. J. F. and Laugier, C. (1997). Parameter identification for dynamic simulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 1928–1933. 53

Gelder, A. V. (1998). Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2):21–42. 53

Georgii, J. and Westermann, R. (2005). Mass-Spring systems on the GPU. In *Simulation Modeling Practice and Theory*, pages 693–702. 55

Geuzaine, C. and Remacle, J.-F. (2008). *Gmsh Reference Manual*. 41

Gibson, J. J. (1966). *The senses considered as perceptual systems*. London. 46

Gibson, S. F. F. and Mirtich, B. (1997). A survey of deformable models in computer graphics. In *Computer Graphics*. 26, 52

Gortler, S. J. and Cohen, M. F. (1995). Hierarchical and variational geometric modeling with wavelets. In *Proceedings of the 1995 symposium on Interactive 3D graphics*. ACM. 53

Gortler, S. J., Schröder, P., Cohen, M. F., and Hanrahan, P. (1993). Wavelet radiosity. In *Proceedings of SIGGRAPH'93*, pages 221–230. 53

Gosline, A. H., Turgay, E., and Brouwe, I. (2002). Haptic illusions: What you feel isn't always what you get. In *Proceedings of Human Interface Technologies*, pages 19–22. 47

Gosselin, F., Ferlay, F., Bouchigny, S., Mégard, C., and Taha, F. (2010). Design of a multimodal VR platform for the training of surgery skills. In *EuroHaptics*, pages 109–116. 51

Griffith, E. (2010). *Visualizing Cumulus Clouds in Virtual Reality*. PhD thesis, Delft University of Technology. 12

Hairer, E. and GerhardWanner (2004). *Solving Ordinary Differential Equations I: Nonstiff Problems*. Third Edition. Springer, Berlin. 37

Hamza-Lup, F. G. and Sopin, I. (2009). Web-based 3D and haptic interactive environments for e-learning, simulation, and training. In *Web Information Systems and Technologies*, volume 18, pages 349–360. 51

Hauser, K. K., Shen, C., and O'Brien, J. F. (2003). Interactive deformation using modal analysis with constraints. In *Graphics Interface*, pages 274–256. 39, 58, 59, 65, 96

Hauth, M., Etzmuss, O., and Tbingen, U. (2001). A high performance solver for the animation of deformable objects using advanced numerical methods. In *Processdings of Eurographics*, pages 319–328. 75

Hayward, V., Astley, O. R., Cruz-Hernandez, M., Grant, D., and Robles-De-La-Torre, G. (2004). Haptic interfaces and devices. *Sensor Review*, 24(1):16–29. 49

Ho, C.-H., Basdogan, C., and Srinivasan, M. A. (1999). Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects. *Presence: Teleoperators and Virtual Environments*, 8(5):477–491. 68

Ho, H.-N. and Jones, L. A. (2007). Development and evaluation of a thermal display for material identification and discrimination. *ACM Trans. Appl. Percept.*, 4. 48

Holt, P. O., Ritchie, J. M., Day, P. N., Simmons, J. E. L., Robinson, G., Russell, G. T., and Ng, F. M. (2004). Immersive virtual reality in cable and pipe routing:design metaphors and cognitive ergonomics. *Journal of Computing and Information Science in Engineering*, 4(3):161–170. 15

Hover, R., Kosa, G., Szekely, G., and Harders, M. (2009). Data-driven haptic rendering-from viscous fluids to visco-elastic solids. *IEEE Transactions on Haptics*, 2(1):15–27. 57

Hua, J. and Qin, H. (2002). Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *Proceedings of symposium on Volume visualization and graphics*, pages 55–64. 50

Indhumathi, C., Chen, W., and Cai, Y. (2009). Multi-modal VR for medical simulation. *The International Journal of Virtual Reality*, 8(1):1–7. 12

Iwamoto, T., Tatezono, M., Hoshi, T., and Shinoda, H. (2008). Airborne ultrasound tactile display. In *SIGGRAPH 2008 new tech demos*, pages 1–1. ACM. 48

James, D. and Pai, D. (2002). DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Transactions on Graphics (SIGGRAPH 2002)*, 21(3):582–585. 58

James, D. L. and Fatahalian, K. (2003). Precomputing interactive dynamic deformable scenes. *ACM SIGGRAPH*, 22:879–887. 56

James, D. L. and Pai, D. K. (1999). Accurate real time deformable objects. In *Proceedings of Computer Graphics (Siggraph 1999)*, pages 65–72. 28

James, D. L. and Pai, D. K. (2004). BD-Tree: output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3). 36, 37

James, D. L. and Pai, D. K. (2005). A unified treatment of elastostatic contact simulation for real time haptics. In *ACM SIGGRAPH 2005*. 68

Jerabkova, L., Wolter, T. P., Pallua, N., and Kuhlen, T. (2005). Adaptive soft tissue deformation for a virtual reality surgical trainer. In *Proceedings of Medicine Meets Virtual Reality (MMVR 2005)*. 12

Julier, S., Lanzagorta, M., Baillot, Y., Rosenblum, L., Feiner, S., Hollerer, T., and Sestito, S. (2000). Information filtering for mobile augmented reality. In *Proceedings of International Symposium on Augmented Reality (ISAR 2000)*, pages 3–11. IEEE. 12

Jung, B., Hoffhenke, M., and I.Wachsmuth (1998). Virtual assembly with construction kits. In *Proceedings of ASME Design for Engineering Technical Conferences*. 13

Kardi, A. (2007). *Contribution de la réalité virtuelle à l'évaluation de produits dans les phases amonts du processus de conception*. PhD thesis, Université d'Angers. 8

Kaus, B., Y.Y.Podladchikov, and D.W.Schmid (2003). Eulerian spectral/finite-difference method for large deformation modelling of visco-elasto-plastic geomaterials. *Bolletino di Geofisica*, 45:346–349. 28

Kim, J., Kim, H., Tay, B. K., Muniyandi, M., Srinivasan, M. A., Jordan, J., Mortensen, J., Oliveira, M., and Slater, M. (2004). Transatlantic touch: a study of haptic collaboration over long distance. *Presence: Teleoper. Virtual Environ.*, 13(3):328–337. 51

Kim, S., Berkley, J. J., and Sato, M. (2003). A novel seven degree of freedom haptic device for engineering design. *Virtual Reality*, 6:217–228. 48

Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). Libmesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254. 43

Ko, J., Kurdila, A. J., and Rediniotis, O. K. (2000). Divergence free bases and multiresolution methods for reduced-order flow modeling. *AIAA Journal*, 38(12):2219–2232. 55

Kobe, G. (1995). Virtual interiors. *Automotive Industries*, 175(5):52–54. 12

Koçak, U., Palmerius, K., and Cooper, M. (2009). Dynamic deformation using adaptable, linked asynchronous FEM regions. In *Spring Conference on Computer Graphics*. 54

Krysl, P., Lall, S., and Marsden, J. E. (2001). Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal For Numerical Methods In Engineering*, 51(51):479–504. 56

Laycock, S. and Day, A. (2004). A hybrid collision detection approach for the haptic rendering of deformable tools. *Computer Graphics International Conference*, 0:148–155. 70

Lécuyer, A., Burkhardt, J.-M., Coquillart, S., and Coiffet, P. (2001). "boundary of illusion" : an experiment of sensory integration with a pseudo-haptic system. In *Proceedings of IEEE Virtual Reality Conference*, pages 115–122. 47

Lederman, S. J. and Klatzky, R. L. (1997). *Haptic aspects of motor control*, volume 11. Elsevier Science. 47

Liarokapis, F., Mourkoussis, N., White, M., Darcy, J., Sifniotis, M., Petridis, P., Basu, A., and Lister, P. F. (2004). Web-3D and augmented reality to support engineering education. *World Transactions On Engineering And Technology Education*, 3(1):11–14. 12

Lin, M. C., Otaduy, M. A., and Boulic, R. (2008). Virtual reality software and technology. *IEEE Computer Graphics and Applications*, 28:18–19. 12

Liu, Y., Jiao, S., Wu, W., and De, S. (2008). GPU accelerated fast FEM deformation simulation. In *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems*, pages 606–609. 55

Liu, Z.-y. and Tan, J.-r. (2002). Interactive sculpting of product shape based on constraint manipulation in virtual environment. *Journal Of Zhejiang University*, 6(7):733–740. 12

Longstaff, A. (2005). *Neuroscience: Science Writer and Neuroscience Lecturer*. Taylor and Francis. 47

Loock, A., Schömer, E., and Stadtwald, I. (2001). A virtual environment for interactive assembly simulation: From rigid bodies to deformable cables. 17

Loomis, J. M. and Lederman, S. J. (1986). *Tactual perception*. Wiley-Interscience. 46

Louchet, J., Provot, X., and Crochemore, D. (1995). Evolutionary identification of cloth animation models. In *Computer Animation and Simulation '95*, pages 44–54. Springer-Verlag. 53

Lumley, J. L. (1967). The structure of inhomogeneous turbulence. *Atmospheric turbulence and wave propagation*, pages 166–178. 55

Maciel, A., Boulic, R., and Thalmann, D. (2003). Deformable tissue parameterized by properties of real biological tissue. In *IS4TM*, pages 74–87. Springer-Verlag. 53

Mahoney, D. P. (1995). Driving VR. *Computer Graphics World*, 18(5):22–23. 12

Mahoney, D. P. (1997). VR drives chrysler's new car. *Computer Graphics World*, 20(7):61–62. 12

Meyrueis, V., Paljic, A., and Fuchs, P. (2009). D3: an immersive aided design deformation method. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 179–182. ACM. 15

Morooka, K., Chen, X., Ryo, K., Uchida, S., Hara, K., Iwashita, Y., and Makoto, H. (2008). Real-time nonlinear FEM with neural network for simulating soft organ model deformation. In *Proceedings of the 11th International Conference on Medical Image Computing and Computer-Assisted Intervention, Part II*, pages 742–749. Springer-Verlag. 57

Morris, D. (2008). Automatic preparation, calibration, and simulation of deformable objects. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(4):263–279. 53

Müller, M., Dorsey, J., McMillan, L., Jagnow, R., and Cutler, B. (2002). Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM. 58

Nealen, A., Müller, M., Keiser, R., Boxerman, E., and Carlson, M. (2005). Physically based deformable models in computer graphics. *Computer Graphics*, 25(4):809–836. 26, 51, 52

Nesme, M., Faure, F., and Payan, Y. (2006). Hierarchical multi-resolution finite element model for soft body simulation. In *proceedings of 3rd International Symposium on Biomedical Simulation*, pages 40–47. 53

Noma, H., Yanagida, Y., and Tetsutani, N. (2003). The proactive desk: a new force display system for a digital desk using a 2-DOF linear induction motor. In *Proceesings of IEEE Virtual Reality*, pages 217–224. 48

Nurnberger, A., Radetzky, A., and Kruse, R. (1998). A problem specific recurrent neural network for the description and simulation of dynamic spring models. In *Proceedings of IEEE World Congress on Computational Intelligence*, volume 1, pages 468–473. 53

Oden, J. (1972). *Finite Elements of Non-linear Continua*. Dover Publications Inc. 29, 31

Pai, D. K., Doel, K. V. D., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L., and Yau, S. H. (2001). Scanning physical interaction behavior of 3D objects. In *Proceedings of Computer Graphics (SIGGRAPH 01)*, pages 87–96. ACM. 57

Patzak, B. and Bittnar, Z. (2001). Design of object oriented finite element code. *Advances in Engineering Software*, 32(10-11):759–767. 43

Pentland, A. and Williams, J. (1989). Good vibrations: modal dynamics for graphics and animation. *Compter Graphics*, 23(3):207–214. 37

Peterlík, I. (2009). *Haptic Interaction with Non-linear Deformable Objects*. PhD thesis, The Faculty of Informatics, Masaryk University, Czech Republic. 40, 72

Peterlík, I. and Matyska, L. (2007). An algorithm of state-space precomputation allowing non-linear haptic deformation modelling using finite element method. In *Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 231–236. 57, 59

Picinbono, G. and Lombardo, J.-C. (1999). Extrapolation: a solution for force feedback? In *International Scientic Workshop on Virtual Reality and Prototyping*, pages 117–125. 59

Picon, F. (2010). *Interaction haptique pour la conception de formes en CAO immersive*. PhD thesis, Université Paris XI. v, 7, 8, 10, 14, 15

Piekarski, W. and Thomas, B. (2002). Arquake: the outdoor augmented reality gaming system. *Commun. ACM*, 45:36–38. 12

Pokluda, L. and Sochor, J. (2005). Spatial orientation in buildings using models with haptic feedback. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. 51

Pontonnier, C. and Dumont, G. (2009). Inverse dynamics method using optimisation techniques for the estimation of muscle forces involved in the elbow motion. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3:227–236. 15

Pontonnier, C. and Dumont, G. (2010). From motion capture to muscle forces in the human elbow aimed at improving the ergonomics of workstations. *Virtual and Physical Prototyping*, 5(3):113–122. 15

Raposo, A., Santos, I., Soares, L., Wagner, G., Corseuil, E., and Gattass, M. (2009). Environ: Integrating VR and CAD in engineering projects. *IEEE Computer Graphics and Applications*, 29:91–95. 17

Renard, Y. and Pommier, J. (2008). *Getfem++, a Generic Finite Element Library in C++*. 42

Renz, M., Preusche, C., Pötke, M., Kriegel, H.-P., and Hirzinger, G. (2001). Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. In *In Proceedings of Eurohaptics*, pages 149–154.

Roberts, J. C. and Paneels, S. (2007). Where are we with haptic visualization? In *Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 316–323. 50

Robles-De-La-Torre, G. (2006). The importance of the sense of touch in virtual and real environments. *IEEE Multimedia*, 13(3):24–30. 49

Rodriguez, A. N., Jessel, J.-P., and Torguet, P. (2002). A virtual reality tool for teleoperation research. *Virtual Reality Journal*, 6(2):57–62. 11

Saddik, E. (2007). The potential of haptics technoligies. *IEEE Instrumentation and measurement magazine*, 10(1):10–17. 52

Sarti, A., Gori, R., and Lamberti, C. (1999). A physically based model to simulate maxillofacial surgery from 3D CT images. *Future Generation Computer Systems - Special issue on ITIS - an international telemedical information society*, 15:217–221. 28

Schmitz, B. (1995). Great expectations: the future of virtual design. *Computer-Aided Engineering*, 14(10):68–72. 12

Schuemie, M. J., Straaten, P. V. D., Krijn, M., and Mast, C. A. V. D. (2001). Research on presence in virtual reality: a survey. In *proceeding of Cyberpsychology and Behavior*. 11

Sederberg, T. W. and Parry, S. R. (1986). Free-form deformation of solid geometric models. In *SIGGRAPH 86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques (1986)*, pages 151–160. 27

Seiler, M., Spillmann, J., and Harders, M. (2010). A threefold representation for the adaptive simulation of embedded deformable objects in contact. *Journal of WSCG*, 18(1-3):89–96. vi, 54, 96

Seth, A., Su, H.-J., and Vance, J. M. (2006). Sharp: A system for haptic assembly and realistic prototyping. In *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. v, 13

Shabana, A. A. (1990). *Theory of Vibration, Volume II: Discrete and Continuous Systems*. Springer-Verlag, New York, NY. 33, 35, 39

Shewchuk, J. R. (2000). Sweep algorithms for constructing higher-dimensional constrained delaunay triangulations. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 350–359. 41

Shi, H. F. and Payandeh, S. (2008). GPU in haptic rendering of deformable objects. In *HAPTICS: PERCEPTION, DEVICES AND SCENARIOS*, pages 163–168. 55

Si, H. (2006). *TetGen, A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. 41

Si, H. and Gärtner, K. (2005). Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In *Proceedings of the 14th International Meshing Roundtable*, pages 147–163. 41

Srinivasan, M. A. and Basdogan, C. (1997). Haptics in virtual environments : Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393–404. 46, 48

Stam, J. (1997). Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 16:159–164. 58

Stork, A., Thole, C.-A., Klimenko, S., Nikitin, I., Nikitina, L., and Astakhov, Y. (2008). Towards interactive simulation in automotive design. *The Virtual Computer*, 24(11):947–953. 17

Sung, R. C. W. and Corney, J. R. (2001). Automatic assembly feature recognition and disassembly sequence generation. *Journal of Computing and Information Science in Engineering*, 1(4):291–300. 13

Sutherland, I. E. (1965). The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508. 47

Tagawa, K., Hirora, K., and Hirose, M. (2008). Manipulation of dynamically deformable object. In *IEEE EuroHaptics2008*. 57, 63

Tagawa, K., Hirota, K., and Hirose, M. (2006). Impulse response deformation model: an approach to haptic interaction with dynamically deformable object. In *Proceedings of 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE. 56

Tagawa, K., Hirota, K., and Hirose, M. (2009). A data compression method for impulse response deformation model. In *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 428–433. 59, 83

Tching, L., Dumont, G., and Perret, J. (2010). Haptic assembly of CAD models using virtual constraint guidance. In *Proceedings of ASME 2010 World Conference on Innovative Virtual Reality (WINVR 2010)*. v, 13, 14

Terzopoulos, D. and Fleischer, K. (1988). Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *Proceedings of SIGGRAPH'88*. 53

Tiest, W. M. B. and Kappers, A. M. (2007). Haptic and visual perception of roughness. *Acta Psychologica*, 124(2):177–189. 47

Tong, C., Song, A., and Juan, W. (2007). A mass-spring model for haptic display of flexible object global deformation. In *International Conference on Mechatronics and Automation (ICMA 2007)*, pages 2753–2757. 50

Treuille, A., Lewis, A., and Popović, Z. (2006). Model reduction for real-time fluids. In *In Proceedings of ACM SIGGRAPH 2006*. 55

van der Horst, B. J. and Kappers, A. M. L. (2008). Haptic curvature comparison of convex and concave shapes. *Perception*, 37(8):1137–1151. 47

van Krevelen, D. and Poelman, R. (2010). A survey of augmented reality technologies, applications and limitations. *The International Journal of Virtual Reality*, 9(2):1–20. 12

Vo, D. M., Vance, J. M., and Marasinghe, M. G. (2009). Assessment of haptics-based interaction for assembly tasks in virtual reality. *World Haptics Conference*, pages 494–499. 14

Vogt, S., Khamene, A., and Sauer, F. (2006). Reality augmentation for medical procedures: System architecture, single camera marker tracking, and system evaluation. *International Journal Of Computer Vision*, 70(2):179–190. 12

Volkov, S. and Vance, J. M. (2001). Effectiveness of haptic sensation for the evaluation of virtual prototypes. *ASME Journal of Computing and Information Sciences in Engineering*, 1(2):123–128. 13

Wael, A., Sara, F., Saeid, N., and Douglas, C. (2010a). Adaptive automatic deformation basis generation for haptic interaction with physically deformable models. In *Virtual Reality International Conference (VRIC2010)*, Laval, France. vi, 57, 59, 72, 83

Wael, A., Sara, F., Saeid, N., and Douglas, C. (2010b). Data-Based dynamic haptic interaction model with deformable 3D objects. In *Proceedings of the 8th IEEE Interantional Conference on Industrial Informatics*, pages 314–318. IEEE. 57

Wall, S. and Harwin, W. (2000). Quantification of the effects of haptic feedback during a motor skills task in a simulated environment. In *Proceedings of Phantom User Research Symposium'00*. 15

Wang, Z. and Dumont, G. (2010). Real Time Interaction with Deformable Industrial CAD Model through Haptic Interface in VR. In *Proceedings of IDMME-Virtual Concept 2010*, pages 20–23, Bordeaux, France. 72, 77

Weiner, I. B., Freedheim, D. K., and Schinka, J. A. (2003). *Handbook of Psychology: Research Methods in Psychology*. JohnWiley and Sons. 46

Weiss, H., Ortmaier, T., Maass, H., Hirzinger, G., and Kuehnapfel, U. (2003). A virtual-reality-based haptic surgical training system. *Computer Aided Surgery*, 8(5):269–272. 12, 52

Wriggers, P. (2003). Computational contact mechanics. *Computational Mechanics*, 32. 84

Yamaoka, M., Yamamoto, A., and Higuchi, T. (2008). Basic analysis of stickiness sensation for tactile displays. In *Proceedings of the 6th international conference on Haptics: Perception, Devices and Scenarios*, pages 427–436. Springer-Verlag. 48

Yang, X., Feng, Y., Li, T., and Wang, F. (2001). Solving sequential decision-making problems under virtual reality simulation system. *Winter Simulation Conference*, 1:905–912. 16

Yannier, N., Basdogan, C., Tasiran, S., and Sen, O. L. (2008). Using haptics to convey cause-and-effect relations in climate visualization. *IEEE Trans. Haptics*, 1:130–141. 51

Yinghui, C., Jing, W., and Xiaohui, L. (2006). Real-time deformation using modal analysis on graphics hardware. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 173–176. ACM. 91

Zheng, C. and James, D. L. (2011). Toward high-quality modal contact sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, 30(4). 85

Zhou, J., Shen, X., and Georganas, N. (2004). Haptic tele-surgery simulation. pages 99–104. 51

Zhuang, Y. and Canny, J. (1999). Real-time simulation of physically realistic global deformation. In *Sketches and Applications (SIGGRAPH'99)*. 55

Zhuang, Y. and Canny, J. (2000). Haptic interaction with global deformations. In *Proceedings of IEEE International Conference on Robotics and Automation*. 35

Zilles, C. B. and Salisbury, J. K. (1995). A constraint-based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems*. 50

Zorriassatine, F., Wykes, C., Parkin, R., and Gindy, N. (2003). A survey of virtual prototyping techniques for mechanical product development. In *Proceedings of the Institution of Mechanical Engineers*, pages 513–530. 9, 17

ANNEXE 2 (dernière page de thèse)

VU:

Le Directeur de Thèse
(Nom et Prénom)

VU:

Le Responsable de l'Ecole Doctorale

Vu pour autorisation de soutenance

Rennes, le

Le Président de l'Université de Rennes 1

Guy CATHELINEAU

VU après soutenance pour autorisation de publication :

Le Président de Jury,
(Nom et Prénom)